1987

Numerical field model simulation of full scale fire tests in a closed spherical/cylindrical vessel.

Raycraft, Janet K.; Kelleher, Matthew D.; Yang, K. T.

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

NUMERICAL FIELD MODEL SIMULATION
OF FULL SCALE FIRE TESTS IN A
CLOSED SPHERICAL / CYLINDRICAL VESSEL

by

Janet K. Raycraft

December 1987

Co-Advisor	M.D. Kelleher
Co-Advisor	K.T. Yang

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL (If applicable) Code 69	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000		7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

NUMERICAL FIELD MODEL SIMULATION AT FULL SCALE FIRE TESTS IN A CLOSED SPHERICAL/CYLINDRICAL VESSEL

12. PERSONAL AUTHOR(S)
Raycraft, Janet K.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM ______ TO ______	1987, December	241

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Field Model; Fire Simulation; Fire Modeling; Numerical Fire Model; Fires in Closed Vessels;

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Most of the casualties incurred during a fire are due to the smoke generated. An understanding of the way smoke and fire spread during a fire would provide a valuable tool to save lives and minimize damage. The Naval Research Laboratory maintains a full scale test facility called Fire-1. The computer model developed in this thesis is based on the actual geometry of Fire-1 and uses field modeling. It is a three dimensional, finite difference model using primitive variables. The model includes local and global pressure corrections, surface radiation, turbulence, strong buoyancy, and conuugate boundary conditions. Given heat input data, the computer code produces pressure, temperature, density, and velocity fields. Experimental fire tests conducted in Fire-1 are used to validate the computer code. Reasonable agreement in the results has been found. Because of the model's

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Prof. M.D. Kelleher	22b. TELEPHONE (Include Area Code) (408) 646-2530	22c. OFFICE SYMBOL Code 69Kk

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-2

UNCLASSIFIED

#18 - SUBJECT TERMS - (CONTINUED)

Enclosed Fires

#19 - ABSTRACT - (CONTINUED)

ability to account for pressure, temperature and smoke buildup, its envisioned use is to predict fires aboard ships and submarines.

Numerical Field Model Simulation of Full Scale Fire Tests in a Closed Spherical / Cylindrical Vessel

by

Janet K. Raycraft
Lieutenant, United States Navy
B.S., University of Minnesota, 1980

Submitted in partial fulfillment of the requirements for the degrees of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
and
MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1987

ABSTRACT

Most of the casualties incurred during a fire are due to the smoke generated. An understanding of the way smoke and fire spread during a fire would provide a valuable tool to save lives and minimize damage. The Naval Research Laboratory maintains a full scale test facility called Fire-1. The computer model developed in this thesis is based on the actual geometry of Fire-1 and uses field modeling. It is a three dimensional, finite difference model using primitive variables. The model includes local and global pressure corrections, surface radiation , turbulence, strong buoyancy, and conjugate boundary conditions. Given heat input data, the computer code produces pressure, temperature, density, and velocity fields. Experimental fire tests conducted in Fire-1 are used to validate the computer code. Reasonable agreement in the results has been found. Because of the model's ability to account for pressure, temperature and smoke buildup, its envisioned use is to predict fires aboard ships and submarines.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to express my appreciation to my thesis advisors, Dr. Matthew Kelleher of the Naval Postgraduate School and Dr. K.T. Yang of the University of Notre Dame, for their patience and guidance with this work. A very special note of thanks to Dr. H.Q. Yang for his invaluable assistance with the computer code, production runs, and graphical output. Without his help this research could not have been accomplished.

I. INTRODUCTION

A. BACKGROUND

A fire, particulary in a closed space such as a room, can be devastating, especially if it is not contained quickly. The danger of a fire lies not only with the flame, but with the toxic gases and smoke emitted during combustion. When a fire ignites, gases leave the surface and mix with air to burn in a turbulent plume causing a hot layer to form below a room ceiling. Unignited objects are being heated primarily by radiation from the chemically reacting flame gases and incandescent soot and to a smaller degree by actual contact with the hot gases. The rate by which these objects are heated is similar, causing them to ignite approximately at the same time. When this happens, there is a sudden engulfment of the room in flames.

The fire safety procedures in practice today are a result of trial and error. To perfect these life saving procedures, a detailed understanding of the fundamental phenomena such as combustion, heat and mass transfer, gaseous radiation, and the flow of gases must be obtained. With this understanding incorporated into the design of enclosed spaces, it is hoped that the probability of ignition and fire spread is kept relatively low. And should a fire break out, those inside should be warned in ample

time in order to extinguish the fire quickly. The ultimate goal is to keep life and property losses at a minimum.

The phenomenon of a fire brings together heat transfer, thermodynamics, chemistry, and aerodynamics plus a dependence on the geometry of the space in which the fire occurs. To predict the nature of a fire, extensive research is required to find out how fire and smoke spread throughout a closed space. This research can be carried out either by experimental work or by a computer model.

Experimental work has been ongoing in the area of fire research. Because of the many phenomena involved in a fire, attempts to apply scaling laws are difficult. Without these scaling laws, the use of small inexpensive tests can no longer be used to predict what will actually happen in a large scale fire. The alternative is to conduct full scale tests that are expensive and somewhat dangerous. Not only one, but many tests are required to ensure the reliability of the data collected. To test another scenario, the test facility would have to be modified, which is again both time consuming and expensive. The physical limitations of the facility alone would limit the types of experiments that could be performed.

The recent advancement in computer speed and storage capability has led to the ability to solve a system of complex partial differential equations that was difficult to attempt before. The various phenomena of a fire are

approximated by simpler models which become building blocks that can be expanded to eventually model the fire accurately. Present day computer models do give reasonable approximations to what actually happens during experimental fire tests. That is why at present it is still important to verify a computer model with an experimental test. Once verified, a computer model can then be modified to adapt to a number of scenarios in order to screen for the one scenario that is potentially the most dangerous. This scenario can be further explored by an experimental test. This eliminates randomly chosen scenarios to conduct expensive tests. The computer model provides additional information unavailable by experimental means. For example the velocity and temperature fields at various time intervals can be determined and plotted to see how a fire spreads. This can reveal areas that require additional experimental data collection. The computer code will be a very powerful tool in predicting fires in other facilities with different geometries once the code reaches completion.

Two different types of fire modeling procedures have been developed:

1) The modular or zone modeling is based on dividing a compartment into distinct regions or control volumes [Ref 1]. Examples of these are as follows, fire plume, hot upper layer, heating of the wall, etc.. All of the control volumes are then interrelated by means of mass and energy balances across the boundaries. This way the entire field is described at any given time by the thermodynamic/fluid dynamic solution. What actually happens in each individual compartment is not always adequately understood.

2) The differential field equation models, or field modeling, is based on dividing the enclosure into many finite volume elements. These models have a strong reliance on the physics of the fire because the proper differential conservation equations are used to calculate the mass, momentum, energy and smoke concentration with the appropriate initial and boundary conditions being applied. For each small volume of gas, the conservation equations for characteristic properties such as temperature, pressure, density, concentration and velocity are monitored to determine the properties of the field at that time. Physical effects such as turbulence and radiation are easily integrated in this field model, but the overall results will depend on the accuracy of these interactive models.

Field models provide the most detailed information about a fire. This information comes at the expense of requiring a large amount of computer resources. The model must have a large number of cells to obtain satisfactory results which does restrict the ability of present day computers to provide real time simulations.

Prior work in the area of field modeling has revealed many conclusions as to how hot gases and smoke spread. Work done at the University of Notre Dame [Refs. 2,3] involves the study of aircraft cabin fires. In dealing with aircraft cabins, a two dimensional finite difference algorithm was used to modeled turbulent buoyant flows. This program monitored how temperature, smoke concentration, and hot gases vary in seating areas. Another two dimensional field model developed at the University of Notre Dame [Ref 4:pp. 1721-1732] describes transient cooling by natural convection using a fully transient semi-implicit upwind differencing scheme with global pressure correction that provided good

results with experimental data. This was for a square enclosure with one vertical wall cooled and the other three walls insulated.

Within the past few years a great deal of progress has been made on the numerical solution of the set of coupled partial differential equations that govern the natural convection process in enclosures. Field models that have been developed for three dimensional rectangular enclosures [Refs. 5-13], use the finite differencing method because of its relative ease of use and its success in solving nonlinear partial differential equations.

Prior work has also been done in three-dimensional cylindrical coordinate buoyant flows [Refs. 14-20]. Most of the cylindrical cavities deal with horizontal cylindrical annuli with differential temperatures specified at inner and outer cylindrical walls. Numerical studies directly related to a horizontal cylinder with differentially heated ends is given by Smutek, et al. [Ref. 19] for low Rayleigh numbers and by Yang, et al. [Ref. 20] for high Rayleigh numbers.

The stream function-vorticity formulation has been used [Refs. 14-19], to do the numerical calculations on natural convection in various geometries. This method has the advantage of decoupling the pressure terms from the momentum equations, thereby satisfying continuity. It does have a number of shortcomings which include becoming unstable at even moderate Rayleigh numbers. Yang, et al. [Ref. 20]

lists these shortcomings, and explains the advantages of using a primitive variable formulation with arbitrary orthogonal coordinates.

The study of natural convection in a spherical annulus was conducted by Ozoe, et al. [Ref. 21] by utilizing the vorticity-vector potential formulation and the alternating-direction-implicit method for Ra = 500.

The geometry that is modeled in this thesis is a combination of cylindrical and spherical geometries. The method developed by Yang, et al. [Ref. 20], is ideal since it involves using a generalized orthogonal coordinate system that can handle complex geometries. The primitive variable formulation is also more desirable due to its stability. That is why the three dimensional model developed here is an extension of the natural convection model in a horizontal cylinder developed by Yang, et al. [Ref. 20].

Field models involving fires in enclosures have been done for room fires [(Ref. 22], and for a general three dimensional enclosure [Ref. 23]. Baum and Rehm [Refs. 24-27] have done extensive research into fire modeling. They employ time dependent inviscid Boussinesq equations to describe a three-dimensional model of buoyant convection and aerosol dynamics in their study of fire induced flow and smoke coagulation.

In studying fires, radiation must also be included. Lloyd, et al. [Ref. 28] have done a numerical study on one

dimensional, surface, gas and soot radiation. Yang [Ref. 29] extended numerical modeling of natural convection-radiation reactions in multidimensional enclosures. Since an efficient overall computational scheme for gaseous radiation is still lacking, radiation involving a participating medium will not be included in the computer model at this time. Only surface to surface radiation is considered.

The Navy has a special interest in fire research. Fires aboard ships or submarines result in fatalities and numerous injuries, not to mention lost operating days and millions of dollars in damages. The Navy has undertaken an extensive program to improve the understanding of how a fire spreads and to improve the methods of extinguishing a fire quickly. Part of the research ongoing includes testing various fire extinguishing equipment or various fire resistant materials.

B. FIRE-1 TEST FACILITY

In order to understand the spread of fire and smoke, the Naval Research Lab (NRL) has a large test chamber called Fire-1 in which full scale fires can be monitored and recorded. The computer code developed here is designed to simulate fires in this facility. This computer model is a first step in predicting the behavior of an actual fire on board a ship.

The computer model will be verified by the experimental data obtained in Fire-1. It is important to include a brief

description of this facility. A more detailed report of Fire-1 is provided by Alexander, et al. [Ref. 30]). Fire-1 is a large scale pressurizable fire test facility that is composed of a cylindrical midsection with hemispherical endcaps. Both the cylindrical section and the endcaps have a 9.6 ft radius, and the overall length is 46.6 ft. In other words, it is a very large pressure vessel capable of being pressurized to 89.7 psi at 450 F. The test chamber is composed of ASTM 285 Grade C steel, 3/8 in thick. The physical description can be found in Table 1.

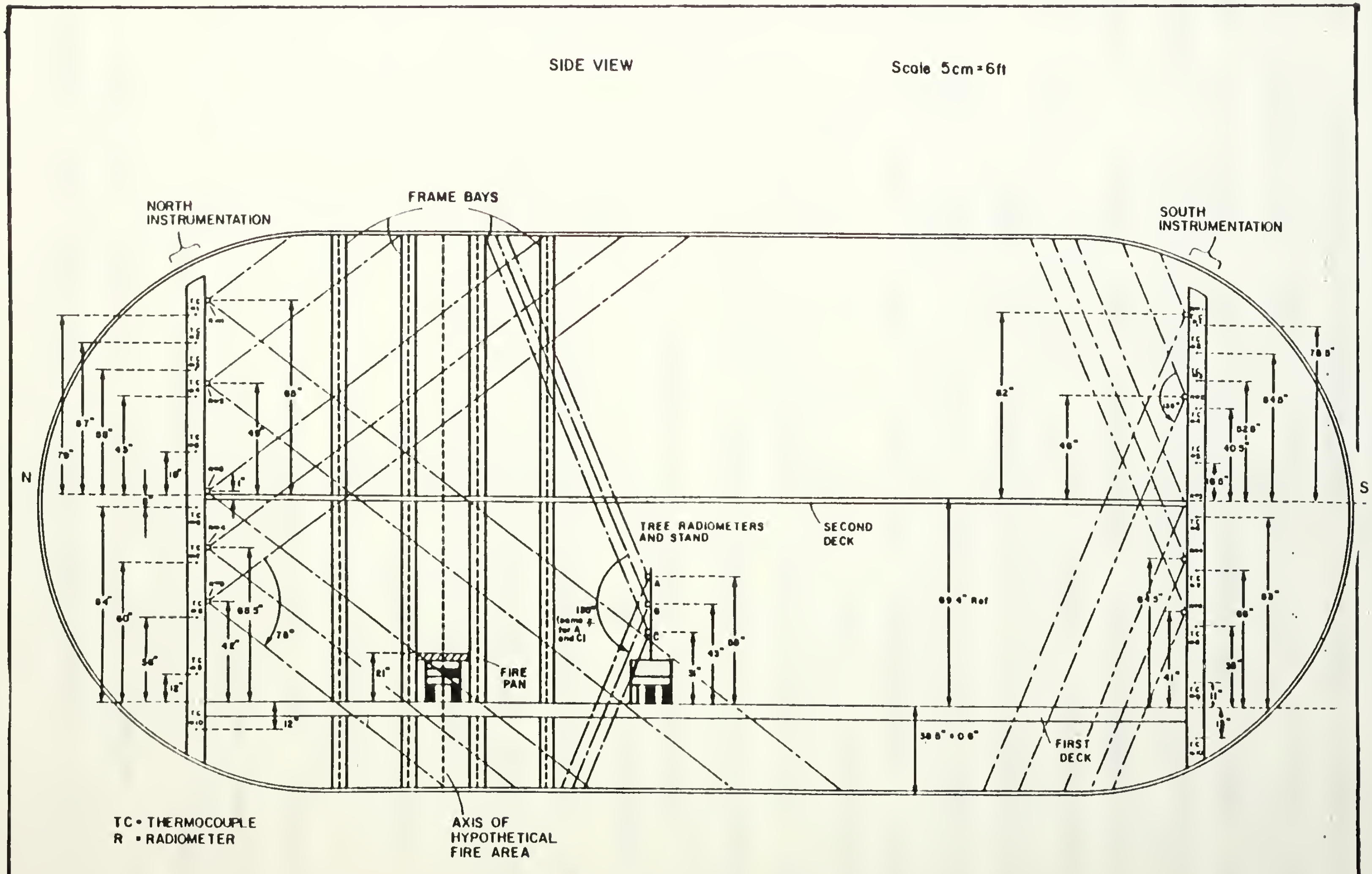
TABLE 1

FIRE-1 TEST FACILITY

Material	3/8" ASTM 285 Grade C Steel
Tank Volume:	
Sphere	3,706 cu. ft.
Cylinder	7,933 cu. ft.
Total	11,639 cu. ft.
Radius	9.6 ft.
Cylinder Length	27.4 ft.
Total Length	46.6 ft.
Pressure Test	ASME Code for 75 psi working pressure internal
Design Pressure	89.7 psia at 450 F
Hydrostatic test	127.2 psia

A fire in Fire-1 is monitored by a number of sensors which include pressure transducers, thermocouples, and radiometers. The test chamber is also instrumented to measure smoke obscuration levels, gas composition and humidity. The use of circulation fans can help to predict what will happen when ventilation is included. A closed circuit television system is also available to record the visual examination of the experiment in progress.

The most important data to the computer model are pressure, temperature, and burn rate. The pressure transducers are located at the north and south ends of the chamber. The temperatures in the chamber are monitored with thermocouple arrays located inside the spherical endcaps as shown in Fig. 1.1. These are chromel-alumel thermocouples with diameters of 0.2 mm and have ceramic insulation enclosed in 304 stainless steel jackets 1.0 mm in diameter. The burn rate is obtained using round, tapered-edge fire pans with various cross-sectional areas, and a constant-level, liquid fuel supply system. The calibration of the system is described by Alexander, et al. [Ref. 30]. Unfortunately, the burn rate data provided up to this point has not been accurate. Another way to obtain this data must be devised or the calibration must be improved as soon as possible, because this data is extremely important in verifying the computer code. Until such time that accurate



— Chamber side view — Thermocouple and radiometer locations in 324 m^3 chamber

Figure 1.1 Side View of Fire I with Sensor Locations

burn rate data becomes available, a method of deducing the burn rate from the pressure data must be used.

The tank has removable steel deck plates which can be solid or an open grate. The horizontal deck can be placed at the midsection, however the deck does not extend into the hemispherical endcaps. The deck is split into two sections over the fire to allow the fire to extend past this second deck and to the overhead. This allows for flexibility in checking the computer code. The first run will verify results in Fire-1 with only the fire present.

C. FIRE-1 COMPUTER MODEL

The Naval Postgraduate School and the University of Notre Dame have undertaken this joint project for the Navy to develop a reliable computer code to predict the spread of fire and smoke in enclosed spaces, whether on board a ship or submarine. This will be used as a powerful tool in the future to assess the effectiveness of proposed damage control systems. It can also be used in the design studies for new ship types. Before this can happen the program has to be verified with simple cases and later modified to include all the complexities of a shipboard fire.

Initial work in this project was conducted by Nies [Ref. 31]. The initial geometry chosen was rectangular, with a volume identical to Fire-1. This was a three dimensional, finite difference model using primitive variables. The model also included global pressure correction, surface

radiation, turbulence, and simple conduction to account for energy losses through the walls of Fire-1. The conclusions he arrived at were:

1. The model predicted expected recirculating flow patterns for the horizontal and vertical planes. This data cannot be recorded at NRL, so the computer model provided additional information.

2. The temperature of the thermocouples located in the upper regions of the spherical endcaps showed significant differences from experimental results. This could be attributed to the geometry difference between the model and the tank.

Since the burn rate data was unavailable, this prevented using the pressure data to validate the computer model. A temporary solution was devised. The scheme artificially developed a heat release curve based on using the experimental pressure curve as an input. From the slope of the experimental pressure curve, a first approximation of the heat input was determined. Initially this guess was fairly good, but as the conduction losses mounted, it became inadequate. The calculated pressure is a function of the heat input, therefore it was used to compute a correction factor. The calculated pressure was compared to the experimental pressure. If it was too large, the heat input was reduced or vice versa. There was a second term in the correction factor to reduce the oscillations by slowing the rate of closure, thereby preventing overshoot. A more detailed account of this procedure is described by Nies [Ref. 31:pp. 61-63]. As Nies noted [Ref. 31] by using this approach, stability problems do arise from taking the

derivative of numerical data and using the fore-mentioned correcting scheme. But until such time that accurate burn rate data can be provided, this is the best method available to attempt a test of the computer model with the data given by NRL.

The model now includes the actual spherical/cylindrical geometry of Fire-1. This required a reformulation of the computer code. The model includes a more detailed formulation of surface radiation, global pressure correction, turbulence and conduction.

The purpose of this thesis is to verify this new model using the spherical/cylindrical geometry by comparing it to the experimental data obtained from Fire-1 with methanol as the fuel burned. Again problems with the inaccuracy of the burn rate data required the computer code to use the elaborate scheme developed by Nies which used the experimental pressure data to deduce a burn rate. In view of the resulting oscillating heat release rates, the results of finite-difference calculations are only used to determine the proper heat release rate input. Consequently, this is taken as trial 1. Another trial, trial 2, was also utilized by inputting a heat release rate curve that corresponded to a curve fit through a set of burn rate data provided by NRL. The burn rate data was taken during the methanol fire run. NRL indicated that the magnitude of the data was possibly off by some unknown scaling factor. The general trend of

the curve seemed reasonable and was used to see how the computer code would predict temperature, pressure and the velocity fields if accurate burn rate data were provided. Results of trial 2 gave an indication of the proper trend of temperature build-up as compared to the experimental data. Based on the combined results of trials 1 and 2, a final trial, trial 3, was then made in the numerical computations to simulate the experimental data as well as to provide the detailed information on the developing temperature and velocity fields.

II. GOVERNING CONSERVATION EQUATIONS

A. GOVERNING EQUATIONS

The computer code developed in this thesis is designed to model Fire-1, the test facility at NRL. As previously described, this facility is a combination of cylindrical and spherical geometries. Prior work in the development of a code to simulate a fire in Fire-1 used a rectangular geometry, Nies [Ref. 31]. The use of a cartesian coordinate system for that simulation was treated as a first approximation. With a spherical/cylindrical geometry, the computer code must be reformulated using a generalized curvilinear coordinate system.

In the development of the equations, various assumptions are made. The fire is modeled by volumetric heat input only. Combustion reactions are not included at this point in time. Density is allowed to vary in accordance with the ideal gas law, and the flow and temperature fields are dominated by turbulent transport.

The governing differential equations are presented in this section along with the transformation from cartesian coordinates into generalized curvilinear coordinates using standard tensor transformation. As Yang, et al. [Ref. 20] pointed out there are several shortcomings that limit the stream function-vorticity formulation procedure to be used

in many applications. From their previous work with flow transitions in three dimensional rectangular tilted enclosures [Refs. 10-12], they have developed a three dimensional primitive variable formulation in arbitrary orthogonal coordinates [Ref. 20]. It is this formulation that is used by the computer model presented here.

1. General Equations

The equations governing the conservation of mass, momentum, energy, and smoke concentration in three dimensional systems can be written in terms of tensor notation as follows:

Continuity

$$\rho_t + (\rho u_i)_{,i} = 0 \tag{2.1}$$

Energy

$$(\rho C_{pm} T)_t + (\rho u_i C_{pm} T)_{,i} = (kT_{,i})_{,i} + \mu\Phi + P u_{i,i} \tag{2.2}$$

Momentum

$$(\rho u_i)_t + (\rho u_i u_j)_{,j} = -P_{,i} - \rho G_i + (\sigma_{ij})_{,j} \tag{2.3}$$

Smoke Concentration

$$(\rho Y)_t + (\rho u_i Y)_{,i} = (DY_{,i,i}) + S_y \qquad (2.4)$$

where ρ is the fluid density, U_i is the velocity vector, the subscript t denotes the derivative with respect to time, P is the static pressure, G_i is the gravity acceleration vector, σ_{ij} is the stress tensor, C_{pm} is the mean isobaric heat capacity, k is the thermal conductivity, μ is the dynamic viscosity, Φ is the dissipation function, Y is the concentration of the smoke, and D is the diffusivity of the smoke. The sheer stress tensor, σ_{ij}, is given by

$$\sigma_{ij} = \mu(u_{i,j} + u_{j,i} - 2/3\ \delta_{ij}u_{k,k}) \qquad (2.5)$$

and the dissipation function is given by

$$\Phi = 2(u_{i,j}^2)\delta_{ij} + [u_{i,j}(1 - \delta_{i,j})]^2 - 2/3(u_{i,i})^2 \qquad (2.6)$$

where the symbol δ_{ij} is the Kronecker delta, which takes on the value 1 when i = j and the value 0 when i = j.

The transformation of these equations into the generalized curvilinear coordinates $(\theta^1, \theta^2, \theta^3)$ is outlined by Yang, et al. [Ref. 20] using the rules in accordance with Eringn [Ref. 32].

The generalized orthogonal coordinates are transformed as

$$X_i \rightarrow \theta^i \qquad (2.7)$$

while a scale factor, h_i, for the curvilinear coordinates in directions θ^i is determined by

$$h_i = (\vec{g}_i \cdot \vec{g}_i)^{1/2} = \left(\frac{\partial X_j}{\partial \theta^i} \cdot \frac{\partial X_j}{\partial \theta^i}\right)^{1/2} \qquad (2.8)$$

Note the summation rule does not apply to the index of h . For cylindrical coordinates, h_1, h_2, and h_3, have the following values [Ref. 33]:

$$h = r = \theta^2 \qquad (2.9)$$

$$h_2 = 1 \qquad (2.10)$$

$$h_3 = 1 \qquad (2.11)$$

For spherical coordinates, the values for h are:

$$h_1 = r \sin\phi = \theta^2 \sin \theta^3 \qquad (2.12)$$

$$h_2 = 1 \qquad (2.13)$$

$$h_3 = r = \theta^2 \qquad (2.14)$$

The covariant metric tensor of orthogonal coordinates is given by

$$g_{ij} = \vec{g}_i \cdot \vec{g}_j = \delta_{ij} h_i h_j \qquad (2.15)$$

which is a special condition since the base vectors are orthogonal and the results are a diagonalized metric tensor. It follows that g is the determinant of g_{ij}

$$g = |g_{ij}| = h_1^2 \; h_2^2 \; h_3^2 \qquad (2.16)$$

The contravariant metric tensor for orthogonal coordinates is determined by

$$g^{ij} = \frac{\delta_{ij}}{h_i h_j} \qquad (2.17)$$

Note the tangent vector to the u_i curve at P is represented by Eqn. 2.18 and the velocity vector is represented by Eqn. 2.19. Both velocity components are in the curvilinear coordinate system.

$$u_i = g_{ij} \; u^{(j)}/h_j \qquad (2.18)$$

$$u^i = u^{(i)}/h_i \qquad (2.19)$$

The generalized orthogonal equations are [Ref. 20]:

Continuity

$$\rho_t + \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^i}(g^{1/2} \rho u^i/h_i) = 0 \qquad (2.20)$$

Energy

$$(\rho C_{pm} T)_t + \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^i} (g^{1/2} \rho C_{pm} u^i T/h_i)$$

$$= \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^i}(g^{1/2} kT_{,i}/h_i^2) + \begin{cases} S_f \\ 0 \end{cases} \qquad (2.21)$$

Smoke Concentration

$$(\rho Y)_t + \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^i} (g^{1/2} \rho u^i Y/h_i)$$

$$= \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^i} (g^{1/2} \rho D Y_{,j} g^{ij}) + \begin{cases} S_{sf} \\ 0 \end{cases} \qquad (2.22)$$

Momentum

$$(\rho u^i)_t + \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^j}(g^{1/2} u^i u^j/h_j) = - P_{,i}/h_i + \rho G^i + \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^j}(g^{1/2} \sigma_i^j/h_j)$$

$$- \frac{1}{h_i h_j} \frac{\partial h_i}{\partial \theta^j}(\rho u^i u^j - \sigma_i^j) + \frac{1}{h_i h_j} \frac{\partial h_j}{\partial \theta^i}(\rho u^j u^i - \sigma_j^j) \qquad (2.23)$$

Stress

$$\sigma_i^j = \mu_{eff} \left[\frac{h_j}{h_i} \frac{\partial}{\partial \theta^i} \left(\frac{u^j}{h_j}\right) + \frac{h_i}{h_j} \frac{\partial}{\partial \theta^j} \left(\frac{u^i}{h_i}\right) + \frac{\delta_{ij}}{h_i h_j} \frac{\partial q_{ii}}{\partial \theta^m} \frac{u^m}{h_m}\right.$$

$$\left. + \frac{\delta_{ij}}{g^{1/2}} \frac{\partial}{\partial \theta^m} \left(g^{1/2} \frac{u^m}{h_m}\right)\right] \tag{2.24}$$

Dissipation

$$\Phi = 2\left[\left(\frac{u^i}{h_i}\right)^2_{;j}\right] \delta_i^j + \left[(u^i/h_i)_{;j}(1 - \delta_i^j)\right]^2$$

$$- 2/3 \left[(u^i/h_i)_{;i}\right]^2 \tag{2.25}$$

The equations are analogous to the cartesian coordinates, except in momentum where two additional terms appear due to Coriolis and centrifugal forces. The definition of stress is also different.

Some terms in the energy equation are combined to form the heat source term, S_f:

$$S_f = \mu\Phi + P \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^i} (g^{1/2} u^i/h_i) \tag{2.26}$$

Since the effects of gas radiation are not treated here, the heat source term is non-zero only in the region of the fire.

B. INITIAL AND BOUNDARY CONDITIONS

In order to solve the governing equations, the initial and boundary conditions must either be given or assumed.

1. Initial Conditions

The initial conditions occur at time equal to zero. This occurs just prior to ignition of the fire in Fire-1. It is assumed there exists a uniform temperature distribution with all the temperatures equal to the ambient temperature. The pressure and density distributions are the static equilibrium distributions in the tank, and the velocity field is set equal to zero to avoid any motion.

2. Boundary Conditions

At any solid boundary in the tank, the velocity components on the wall are set equal to zero due to the no slip conditions. Since the velocity normal to any surface is zero, so is the mass flux. Also the temperature of the solid is equal to the temperature of the fluid at these interfaces.

$$u^i = 0 \tag{2.27}$$

$$T_s = T_{f\ell} \tag{2.28}$$

$$\frac{\partial Y_i}{\partial n} = 0 \tag{2.29}$$

where n is the inward normal.

At the solid boundary, continuity of heat flux must be satisfied.

$$q_r - k_f \frac{\partial T}{\partial n} = - k_s \frac{\partial T_s}{\partial n} \quad (2.30)$$

where q_r is the thermal radiation energy. At the exterior wall, heat is convected away.

Special treatment must also be given for the singularity at r equal to zero for the cylindrical coordinate system. Yang, et al. [Ref. 20:pp. 167-168] explained the different approaches that have been made to rectify this problem, but they chose to use two consecutive radial control volumes placed in the vicinity of r equal to zero. Trying a number of methods, they found this gave the best representation for the temperature and flow fields. It is this approach that is utilized here.

III. RADIATION MODEL

A. INTRODUCTION

In order to calculate the radiation effects in the model, a number of assumptions have to be made. First only surface radiation effects are considered. This means that the gas inside of the tank is modeled as nonparticipating and transparent. This assumption will lead to an increase in the heat transfer to the vessel walls [Ref. 28:pp. 142-164] and the energy equation at the walls of the tank will have to be modified to account for the direct deposit of energy from the fire. The second assumption is that the surfaces are grey and the radiation reflected or emitted from any surface is diffusely distributed. The third assumption defines what is a surface. Both the tank wall and the flame are modeled as a specified number of cells each small enough to be considered as a differential zone.

B. THE METHOD FOR CALCULATING THE RADIANT HEAT TRANSFER

The radiation model is based on the net radiosity method as outlined in Sparrow and Cess [Ref 34:pp 90-94] and summarized here.

In an enclosure, the net rate of heat loss, Q, from a typical surface "i" is the difference between the emitted

radiation and the absorbed portion of the incident radiation.

$$\frac{Q_i}{A_i} = \varepsilon_i \sigma T_i^4 - \alpha_i H_i \tag{3.1}$$

where σ is the Stefan-Boltzmann constant, ε_i is the emissivity, α_i is the absorptivity, and H_i is the radiation incident on surface i per unit time and unit area.

In order to simplify the above equation, a number of assumptions must be made. The tank represents an enclosure composed of N finite surfaces. Each surface is assumed to be isothermal. The participating surfaces are gray, that is, the emitted and the incident radiation are independent of wavelength. From Kirchoff's Law:

$$\alpha_i = \varepsilon_i \tag{3.2}$$

The radiation reflected and emitted from any surface is diffusely distributed. This will simplify the analysis since the radiant energy streaming away from a surface is the sum of the emitted and reflected radiation. Since they are both diffusely distributed, then they are directionally indistinguishable and there is no need to treat them separably. Since H represents the incident radiant energy arriving at a surface, ρH would be the fraction of energy that is reflected from the surface. The total radiant

energy that streams away from a surface is termed the radiosity and is denoted by the symbol B.

$$B = \varepsilon\sigma T^4 + \rho H \qquad (3.3)$$

The radiosity is composed of the radiation emitted and reflected by the surface. It is also assumed that the radiosity of any surface is uniform along that surface. Upon eliminating the flux from Eqn. 3.3, and applying Eqn. 3.2, the following equation is obtained.

$$\frac{Q_i}{A_i} = \frac{\varepsilon_i}{1 - \varepsilon_i} (\sigma T_i^4 - B_i) \qquad (3.4)$$

For an opaque material, the incident radiation is either absorbed or reflected, that is,

$$\alpha + \rho = 1 \qquad (3.5)$$

$$\rho = 1 - \alpha \qquad (3.6)$$

From Eqn. 3.2 and Eqn. 3.6, the radiosities are found by applying Eqn. 3.3 at each of the surfaces in the enclosure.

$$B_i = \varepsilon_i \sigma T_i^4 + (1 - \varepsilon_i) H_i \qquad (3.7)$$

The radiant flux H_i is formed by the summation

$$H_i = \sum_{j=1}^{N} B_j F_{Ai-Aj} \tag{3.8}$$

where B_j is the radiosity at surface "j" and F_{Ai-Aj} is the view factor from surface "i" to surface "j". The radiosity at surface "i" now becomes

$$B_i = \varepsilon_i \sigma T_i^4 + (1 - \varepsilon_i) \sum_{j=1}^{N} B_j F_{Ai-Aj} \qquad 1 \le i \le N \tag{3.9}$$

In this way there are generated N linear, inhomogeneous, algebraic equations for N unknown radiosities. By solving the simultaneous linear algebraic equations, B can be found and then the heat transfer rates Q .

This solution, however, needs to be resolved many times when transient operating conditions are being analyzed for the enclosure. A better way to handle the solution is to find a direct relationship between unknown heat fluxes and prescribed temperatures. Equation 3.9 is rephrased as

$$\sum_{j=1}^{N} X_{ij} B_j = \Omega_i \qquad 1 \le i \le N \tag{3.10}$$

where

$$X_{ij} = \frac{\delta_{ij} - (1 - \varepsilon_i) F_{Ai-Aj}}{\varepsilon_i} \tag{3.11}$$

$$\Omega_i = \sigma T_i^4 \tag{3.12}$$

Evaluating the equation for $i = 1,2,...N$, an N by N array is formed and will designated by matrix X. A column vector of radiosities and temperatures raised to the fourth power for surfaces $i = 1\text{-}N$ is designated B and T^4 respectively.

The system of equations can now be represented by

$$[X] \langle B \rangle = \sigma \langle T^4 \rangle \tag{3.13}$$

To find radiosity, the inverse of X is multiplied by both sides of the equation.

$$\langle B \rangle = \sigma [X]^{-1} \langle T^4 \rangle \tag{3.14}$$

This can now be substituted into Eqn. 3.4.

$$\frac{Q_i}{A_i} = \sum_{j=1}^{N} G_{ij} \sigma T_j^4 \tag{3.15}$$

where

$$\psi_{ij} = X_{ij}^{-1} \tag{3.16}$$

$$G_{ij} = \frac{\varepsilon_i}{1-\varepsilon_i} (\delta_{ij} - \psi_{ij}) \tag{3.17}$$

In this variation of the equation, ψ_{ij} only depends on emittances which are regarded as constants and do not depend

on the temperature. Once the temperatures are known, the heat flux of the surfaces can be calculated.

C. VIEW FACTOR CALCULATIONS

The view factor (alternatively defined as the angle factor, shape factor or geometrical factor) provides information on the fraction of radiant energy leaving one surface that arrives at a second surface.

Sparrow and Cess [Ref. 34:pp. 120-125] provide the general definition of the shape factor.

$$F_{Ai-Aj} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \beta_i \cos \beta_j \, dA_i \, dA_j}{\pi r^2} \qquad (3.18)$$

In this equation the subscript "i" indicates the surface from which the radiation is leaving from and the "j" indicates the surface to which the radiation is going. The length of the connecting line between the two elements is r, and the angles β_i and β_j are formed by the respective surface normals and r.

Due to the geometries associated with the tank, the equations to evaluate the view factors are not found in the literature. This presents a problem that can be solved in one of two ways. The tank could be divided into a few finite areas resulting in the evaluation of the integral over a complex area. Or the tank could be divided into a number of smaller areas that could be assumed small enough to be considered differential in size.

It was decided to use the latter method. Equation 3.18 can be used with the assumption if the areas are small, then the integrand is assumed constant. This avoids any integration.

$$F_{Ai-Aj} \approx dF_{dAi-dAj} = \frac{\cos \beta_i \cos \beta_j}{\pi r^2} dA_j \tag{3.19}$$

This is a reasonable assumption since the error introduced calculating the view factors in this manner is of the same order of magnitude as that of the finite difference algorithm. In order to find the view factor from surface "j" to surface "i", reciprocity can be used since the leaving radiant fluxes are diffusely and uniformly distributed. Sparrow and Cess showed [Ref. 34] that the view factors depend only on the geometrical orientation of the participating surfaces for isothermal, gray, diffuse surfaces. The following equation is then used.

$$A_i F_{Ai-Aj} = A_j F_{Aj-Ai} \tag{3.20}$$

The first consideration in developing the view factors for the tank with the fire present was to find the view factors between elements on the walls of the tank alone, then the effect of the fire would be added along with the effects of shading. The tank is divided into 560 cells, 100 on each endcap and 360 on the cylinder. Each cell is now a

surface radiation zone. The previous rectangular geometry used by Nies [Ref. 31] had 66 surface radiation zones. The surfaces on the tank are slightly concave, but are assumed to be flat to avoid any self radiation. This is a valid assumption due to the small size of the cells and the minimal amount of radiation that would be reflected back on the same cell.

In order to have a means by which the program for calculating view factors can be checked, a useful property of the view factors is deduced from the energy conservation principal. As stated in [Ref. 34:p. 83], the radiant energy leaving any surface in an enclosure must impinge on any other surface in the enclosure whereby none can be lost.

This leads to the following equation.

$$\sum_{j=1}^{N} F_{Ai-Aj} = 1 \tag{3.21}$$

The N denotes the number of surfaces in the enclosure.

1. <u>Tank Element to Tank Element View Factors</u>

There are three general types of view factors and their reciprocals associated with the tank.

a) spherical element to spherical element
   - same hemisphere
   - opposite hemispheres
b) spherical element to cylindrical element
c) cylindrical element to cylindrical element

To illustrate the process by which these view factors were found refer to Fig. 3.1 which shows how the cylindrical element to spherical element view factors were obtained. The other view factors were found in a similar fashion. First the distance between the two elements was obtained by using relations for right triangles.

$$a^2 = R^2 + \rho^2 - 2\rho R \cos\theta \tag{3.22}$$

$$b^2 = (\Delta z + h)^2 \tag{3.23}$$

$$r^2 = a^2 + b^2 \tag{3.24}$$

Next the cosine of the angle between the normal of the element and the distance r must be found. For element 1, ζ_1 is found in the following manner then cosine β_1

$$\zeta_1^2 = b^2 + \rho^2 \tag{3.25}$$

$$\cos\beta_1 = \frac{R^2 + r^2 - \zeta_1^2}{2Rr} \tag{3.26}$$

For element two a similar analysis is made.

$$\zeta_2^2 = \Delta z^2 + R^2 \tag{3.27}$$

$$\cos\beta_2 = \frac{R^2 + r^2 - \zeta_2^2}{2Rr} \tag{3.28}$$

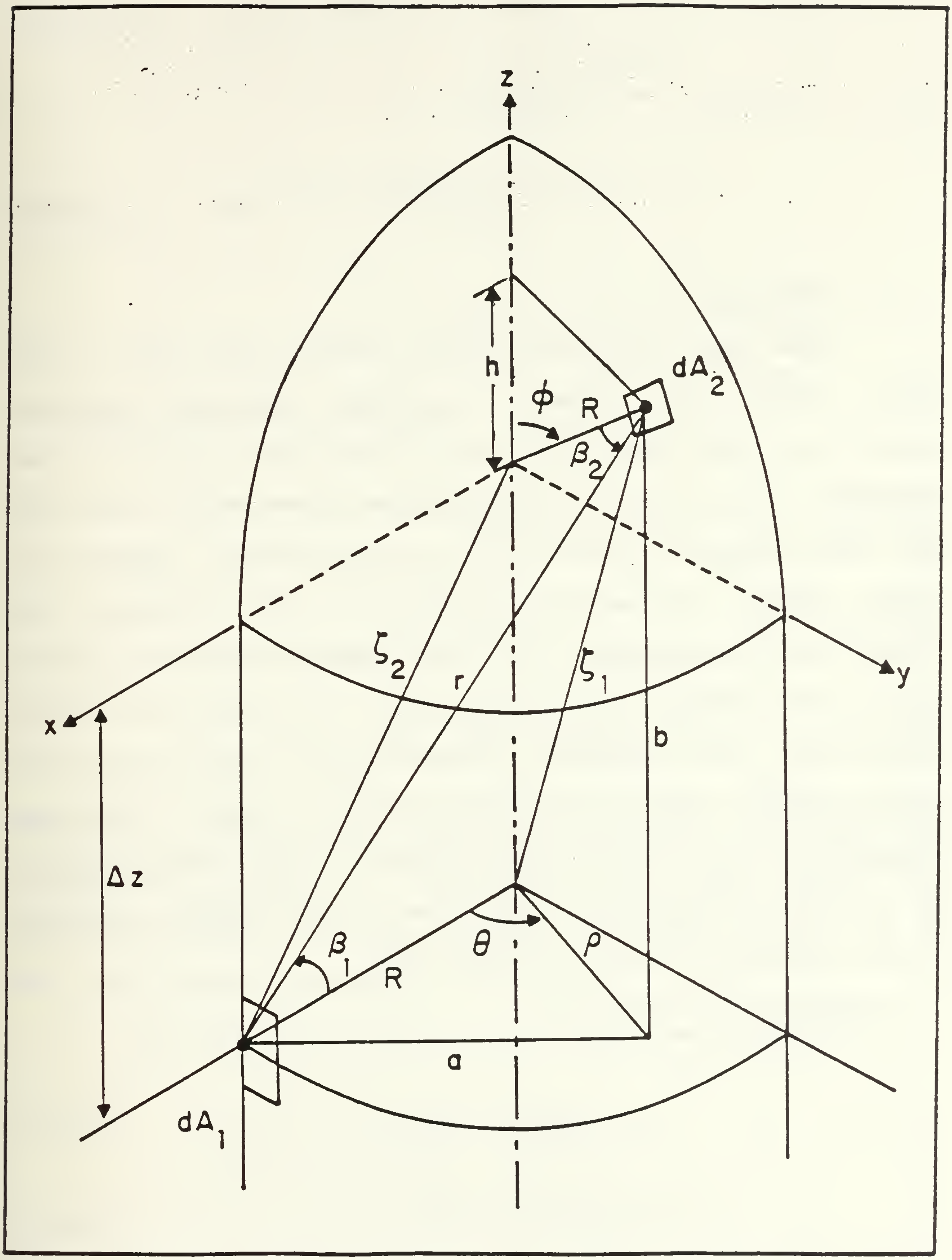


Figure 3.1 Calculation of Cylinder to Sphere View Factor

which results in the view factor being:

$$dF_{A1-A2} = \frac{\cos \beta_1 \cos \beta_2}{\pi r^2}$$

$$= \frac{(r^2 - \Delta Z^2)(R^2 + a^2 - \rho^2)}{4\pi R^2 r^4} dA_2 \qquad (3.29)$$

2. <u>Shading</u>

The next consideration was putting the fire in the tank. Once this happens, a problem concerning shading enters in. The fire will lie in the direct path of some elements. The elements that would be affected are those on the north sphere to those on the south sphere, elements on either sphere to certain elements on the cylinder, and elements on the cylinder to other elements on the cylinder but on the opposite side of the fire. If the line of site between any two elements intersected the fire, the view factor between the two elements was set to zero. This was accomplished in the following manner. First the equation for the line of sight was determined. Each element is given a x,y,z location in the following way:

- spherical to cartesian

$$X = R \sin \phi \cos \theta \qquad (3.30)$$

$$Y = R \sin \phi \cos \theta \qquad (3.31)$$

North End

$$Z = R \cos \phi \tag{3.32}$$

South End

$$Z = Zcyl2 - R \cos \phi \tag{3.33}$$

- cylindrical to cartesian

$$X = R \cos \theta \tag{3.34}$$

$$Y = R \sin \theta \tag{3.35}$$

$$Z = Zcyl1 + Z_c \tag{3.36}$$

where zcyl1 and zcyl2 are specific locations on the tank and illustrated in Fig. 3.2 and Z_c is the length along the cylinder portion of the tank only. The z axis origin is at the north end of the tank.

The equation of the line between the elements is:

$$\frac{X - X_i}{X_j - X_i} = \frac{Y - Y_i}{Y_j - Y_i} = \frac{Z - Z_i}{Z_j - Z_i} = t \tag{3.37}$$

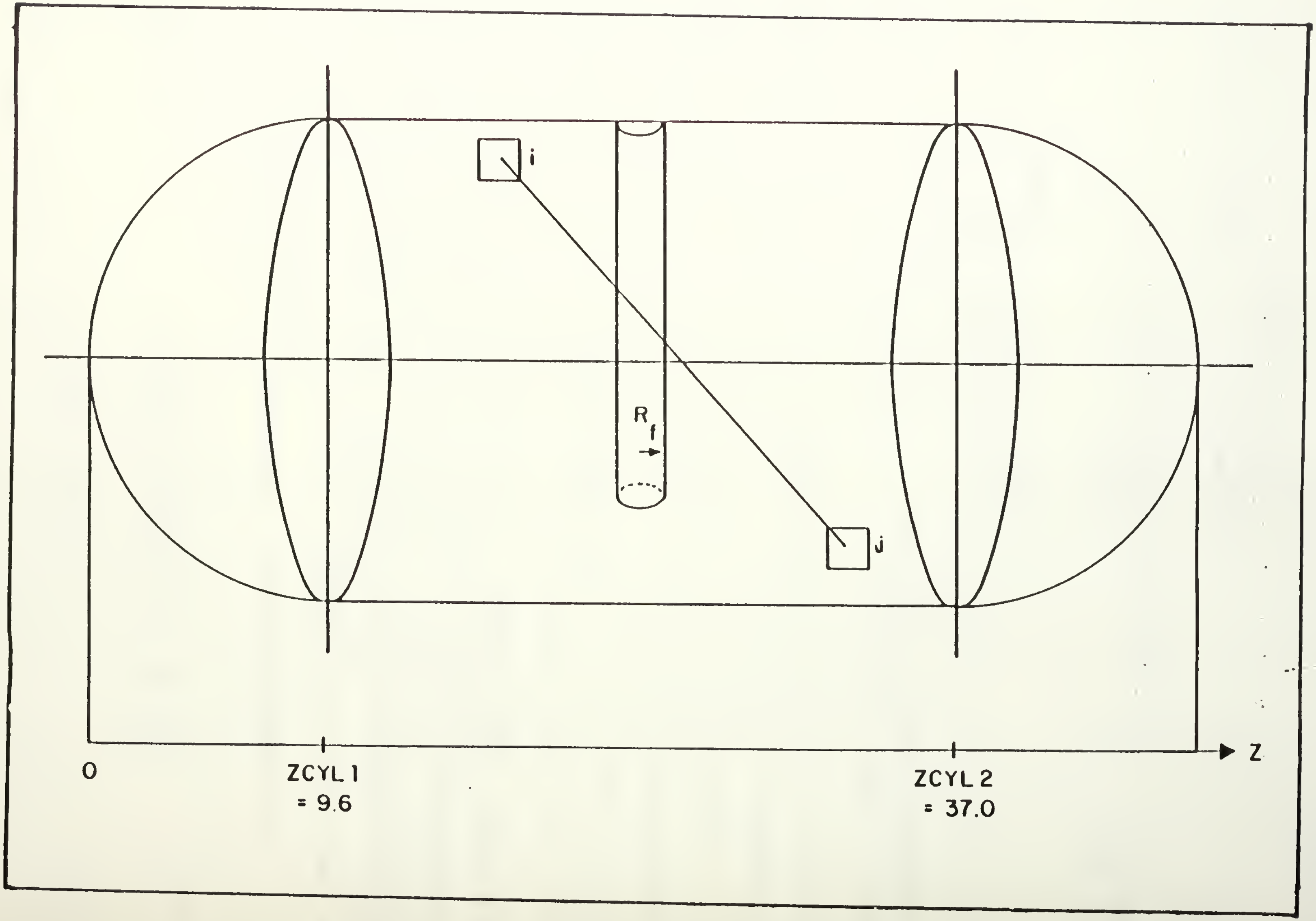


Figure 3.2 Illustration of Shading

The equation for the fire is:

$$X^2 + (Z - HSZ)^2 = R_f^2 \tag{3.38}$$

$$(-y_f \le y \le R)$$

where rf is the radius of the fire, and HSZ is the z location of the vertical axis of the fire centerline.

To find the intersection, substitute Eqn. 3.37 into Eqn. 3.38.

$$At^2 + Bt + C = 0 \tag{3.39}$$

where

$$A = (X_j - X_i)^2 + (Z_j - Z_i)^2 \tag{3.40}$$

$$B = 2X_i(X_j - X_i) + 2(Z_j - Z_i)(Z_i - HSZ) \tag{3.41}$$

$$C = X_i^2 + (Z_i - HSZ)^2 - R_f^2 \tag{3.42}$$

If the following is true, there is no shading and the view factor remains unchanged.

$$B^2 - 4AC < 0 \tag{3.43}$$

Otherwise two solutions for t can be found which result in two solutions for y, y1 and y2, from Eqn. 3.37. The solutions for y must now be checked to see if they coincide with the fire.

$$\text{If} \quad - y_f < \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} < R \tag{3.44}$$

If this equation is true, then there is shading and the view factor between the two elements must be set equal to zero.

3. <u>Fire Element to Tank Element View Factors</u>

The fire is modeled as a vertical cylinder with the diameter equal to that of the experimental fire pan. The fire pan rests on a small deck in Fire-1, the fire in the computer model then extends from the same location to the top of the tank. The fire height can be modified if required. In order to calculate the view factors, it is assumed that the fire can be divided into 19 equal sized cells. The midpoints of these cells lie on the verical axis of the cylinder. When actually calculating the view factors, the tank cells have a line of sight to the midpoint of a fire cell that lies in a two dimensional plane facing the tank cell. This further models the fire as if a plane extends through the axis and now rotates around the axis to face the desired tank cell. The area the tank element sees can actually be rectangular, circular, or a combination of

the two. This unknown area will ultimately cause problems with the calculations and a modification factor will be added.

In calculating the heat source view factors, a number of problems were encountered and solved by making modifications or assumptions. These problems were as follows.

- two sidedness of the fire cells
- geometry of the fire cells close to the tank wall
- actual area of the fire cell.

The first problem is a result of the modeling of the fire. As the plane rotates about the axis, it sees all 560 cells. It is better to model the plane as one side seeing only cells on the north end of the tank and the other side of the plane seeing only the south tank cells. When this is done, the enclosure property can be used to check the accuracy of the view factor calculation. If this modification is not done, then the sum from the enclosure property would be two vs. one.

When calculating the view factors from the fire cells to all other cells in the tank, the enclosure property was within tolerence for those fire cells in the center of the tank, but the closer the cell was to the tank wall, the further the total summation deviated from 1.0. The view factor equation used, Eqn. 3.19, represents the radiant energy leaving dA_i that is incident on dA_j. It was derived

using the relations between the radiosity, intensity, and the solid angle subtended by dA_j when viewed from dA_i. The solid angle, is represented by

$$d\omega = dA_j \cos \beta_j / r^2 \tag{3.45}$$

As the fire cells become extremely close to the tank cell, this solid angle can not be approximated by assuming infinitesimal areas. The initial assumptions used to calculate the view factors do not give a totally accurate result in this case. The view factors are underestimated, resulting in the total summation being less than one. The cells that are most effected by this, are those tank cells directly over the fire. The angle, β_1, between the fire cell normal and the line between the fire cell and these cells is almost 90 degrees. When this happens the greatest modification is required. As this angle goes to zero, the tank and fire cells can accurately use the assumptions stated in the beginning of this section, for the solid angle subtended is very small.

A global modification routine was used to calculate the view factor from the heat source to the tank. First the sum of the view factors from the fire to the tank walls must equal one from the enclosure property. As stated before, this is for each side of the fire. Since β_1 is the

important parameter, a total modification factor, H, is found using β_1.

$$H = f(1 - \cos \beta_1) \tag{3.46}$$

where f is a modification factor unique to each fire cell. The new view factor can be calculated from the old view factor by the following equation.

$$VF_{(new)} = VF_{(old)}\{1 + H\} \tag{3.47}$$

Before this equation can be used, f must be found. Since the sum of the view factors must be one, then the sum of the right hand side of Eqn. 3.47 must also equal one.

$$\sum_{j=1}^{N} VF_{old} + \sum_{j=1}^{N} VF_{old} \cdot H = 1 \tag{3.48}$$

$$\sum_{j=1}^{N} VF_{old} + \sum_{j=1}^{N} fVF_{old} \; (1 - \cos \beta_j) = 1 \tag{3.49}$$

Solving for f

$$f = \frac{1 - \sum_{j=1}^{N} VF_{old}}{\sum_{j=1}^{N} VF_{old} \; (1 - \cos \beta_j)} \tag{3.50}$$

This f is calculated for every fire cell, after the initial view factors are found. The closer the fire cell is to the

tank cell, the larger f would become. Once f is found, each view factor from the fire cell to the tank cell would be modified using Eqn. 3.50. The view factors calculated using reciprocity would also be modified.

To further refine the process, a second iteration was used to find a new f, and then this new f was used to improve the view factors for a second time. Two iterations were all that was required to obtain excellent results with the total summation from the fire cells to all cells either on the north end or the south end being extremely close to one.

The last problem was the summation from one tank cell to all other cells in the enclosure, whether they be other tank cells or fire cells, was not equal to one. If the fire was not included, the summation was equal to one. Which indicated the problem was with the reciprocity relation between the fire cells and the tank cells. In calculating the view factors from the fire to the tank, the exact area of the tank cells was known. But using reciprocity, an assumption had to be made for the area of the fire cell. The first assumption had the area of the fire being a rectangle, a circle or a combination of the two depending on the angle β_1.

$$A_{FC} = A_R(1 - \cos \beta_1) + A_C(\cos \beta_1) \qquad (3.51)$$

where A_{FC} is the area of the fire cell, A_R is the area of the rectangle, and A_C is the area of the circle.

Another modification was in order for the tank cells to the fire cells. The first step was to calculate a modification factor for each tank element.

$$A_{(i)} = \frac{[1 - \sum_{j=1}^{N} VF(i,j)]}{\sum_{j=561}^{579} \frac{VF(i,j)}{SQRT[1 + (R(i,j)/R_f)^2]}} \qquad (3.52)$$

where the numerator involves the difference between the desired value of one using the enclosure property and the actual value obtained from summing all the view factors in the enclosure. The denominator involves a summation only over the nineteen fire cells, where R(i,j) is the distance between the tank and fire cell and Rf is the radius of the fire pan.

The new view factor is then calculated by the following equation.

$$VF_{new}(i,j) = [1 + \frac{A(i)}{SQRT[1 + (R(i,j)/R_f)^2]}] *VF_{old}(i,j) \qquad (3.53)$$

Again this modification is only applied on the view factors from the tank to the fire, i can vary from 1-560 and j can vary from 561 to 579. The sum from any tank cell to all other cells in the enclosure is now found to be one.

The view factors from any fire cell to any other fire cell were set equal to zero since all cells were assumed to lie on a vertical line and not allowed to see each other.

IV. OTHER PHYSICAL MODELS

Besides including radiation into the field model, other aspects of the fire must also be considered. It is the purpose of this chapter to briefly outline the other models incorporated into the program at this time.

A. CONDUCTION MODEL

The computer model must account for the loss of heat by conduction through the tank walls. In the later stages of the fire this becomes increasingly important.

A simple conduction model is proposed here. The model employs one dimensional, unsteady conduction through the tank wall thickness. Convection at the exterior wall is modeled with a constant heat transfer coefficient. The energy equation as applied to the solid wall becomes:

$$(\rho_s C_{ps} T)_t = \frac{1}{g^{1/2}} \frac{\partial}{\partial \theta^i} (g^{1/2} k_s T_{,j} g^{ij}) + S \tag{4.1}$$

where $\rho_s C_{ps}$ is the heat capacitance of the wall and k_s is the wall conductivity.

B. TURBULENCE MODEL

The turbulence model used in the program is a simple algebraic model. The algebraic model can adequately predict the average values of the dependent variables.

The effective viscosity, μ_{eff} , in recirculating buoyant flows with large variations in turbulent level was modeled by Nee and Liu [Ref. 35]. Applying the transformation to a generalized orthogonal coordinate system, the equation can now be written as

$$\frac{\mu_{eff}}{\mu_o} = 1 + \frac{[(\frac{1}{h_j}\frac{\partial u^i}{\partial \theta^j})^2(1-\delta_i^j)]^{1/2}(\frac{\ell}{H})^2}{2+\frac{Ri}{Pr_t}} \tag{4.2}$$

where Ri is the Richardson Number:

$$Ri = \frac{H}{U_o^2}\,\frac{(\frac{\partial T}{\partial n})\vec{n}\cdot\vec{g}}{[(\frac{\partial u^1}{\partial n})\vec{n}\cdot\vec{g}]^2+[(\frac{\partial u^2}{\partial n})\vec{n}\cdot\vec{g}]^2+[(\frac{\partial u^3}{\partial n})\vec{n}\cdot\vec{g}]^2} \tag{4.3}$$

ℓ/H is the non-dimensional mixing length parameter defined as

$$\frac{\ell}{H} = K\left\{\frac{(u^i u^i)^{1/2}}{[\sum_{i,j}(\frac{1}{h_j}\frac{\partial u^i}{\partial \theta^j})^2]^{1/2}} + \frac{[\sum_{i,j}(\frac{1}{h_j}\frac{\partial u^i}{\partial \theta^j})^2]^{1/2}}{[\sum_{i,j}(\frac{1}{h_i h_j}\frac{\partial^2 u^1}{\partial \theta^i \partial \theta^j})^2]^{1/2}}\right\} \tag{4.4}$$

Pr_t is the turbulent Prandtl number, n is a unit vector in the negative gravity direction, and K is an adjustable constant.

The effective conductivity is found by the following equation

$$k_{eff} = \frac{1}{Pr} + \frac{1}{Pr_t} \frac{\mu_{eff}}{\mu_o} \tag{4.5}$$

Pr is the molecular Prandtl number.

V. FINITE DIFFERENCE CALCULATIONS

A. INTRODUCTION

In the formulation of the computer model, the governing differential equations must be modified in order to use numerical methods to solve for the primitive variables. The independent variables are space and time. The dependent, or primitive, variables are the velocity components u^1, u^2, u^3, pressure P, temperature T, and density ρ. The six equations required to solve for these unknowns are the conservation equations of mass, momentum, and energy plus the equation of state. The conservation equations were developed in Chapter II (Eqns. 2.20, 2.21, 2.23). The equation of state for a perfect gas is:

$$P = \rho RT \tag{5.1}$$

where R is the universal gas constant.

In Patankar's book [Ref. 36:pp 25-40], he describes the discretization concept as it applies to a finite difference method. The general form of the finite difference equations for the computer model developed in this thesis follows Doria's initial work [Ref. 37] done at the University of Notre Dame. Doria [Ref. 37:pp. 1-44] discretized the governing equations using a control volume method which was

one of the methods described by Patankar [Ref. 36]. In this method, the flow domain is divided into many separate control volumes. The conservation equations are written for each cell in integral form. This will lead to a set of finite difference equations.

The control volume approach uses the integral form of Eqns. 2.20, 2.21, and 2.23. All properties are at the center of the cell and represent the overall average values. If a numerical method violates the conservation property, non-physical results may be obtained due to artificial sources or sinks of mass, momentum, or energy. When primitive variables are used vice the stream vorticity method, special procedures are needed to handle the pressure coupling among the equations. An iteration procedure is used to estimate pressure. The pressure is then corrected to ensure mass is conserved at each cell. A local pressure correction procedure is discussed by both Patankar [Ref. 36: pp. 120-126] and Doria [Ref. 37:pp. 26-32]. A global pressure correction must also be included in the model to handle net energy changes in the system. Nicolette, et al. [Ref. 4] describes this procedure.

The finite difference equations are solved by an iterative solution procedure. For a nonlinear problem, it is not necessary or practical to take the solution of the algebraic equations to final convergence for a fixed set of coefficient values. Various schemes have been developed

over the years in attempt to obtain a finite difference solution of the flow problem. The central difference solution was found to be unsuitable because it gave a physically unreal oscillatory behavior in simulations where convection is important. The upwind differencing scheme takes into account the unsymmetrical phenomenon of convection by using backward differencing in the direction of flow for the first order derivative. The upwind scheme gives physically reasonable results for most grid Peclet numbers and is free of numerical oscillations. Severe errors do enter in at small grid Peclet numbers because of truncation errors. The upwind differencing scheme also overestimates diffusion at high Peclet numbers. Both schemes can be improved by reducing the grid Peclet number which implies reducing the grid. This is impractical because of limited computer resources.

Another scheme for convective modeling has been developed by Leonard [Ref. 38]. It is called QUICK (Quadratic Upstream Interpolation for Convective Kinematics). It has the accuracy feature of central differencing and retains the stable convective sensitivity of upwind differencing. The discretization equations do not necessarily have a diagonally dominant coefficient matrix and therefore require iteration. H.Q. Yang [Ref. 13] demonstrates the application of the QUICK scheme in coupled

momentum, energy and pressure equation solutions for three dimensional flow in tilted rectangular enclosures.

In the following sections, the control volume method will be applied to the spherical/cylindrical geometry of this computer model. The conservation equations will be integrated. And the finite difference equations will be formed using the QUICK scheme. Further iterative procedures will be added to correct for pressure.

B. CONTROL VOLUME APPROACH

In the control volume method, the total volume to be modeled is divided into a number of nonoverlapping control volumes. In each control volume, or cell, there exists one grid point. A spherical three dimensional cell and its neighbors is illustrated in Fig. 5.1, and a cylindrical cell is illustrated in Fig. 5.2. The cell is centered at a nodal point, P or (I,J,K). The points at the neighboring cells are designated, east (I+1,J,K), west (I-1,J,K), north (I,J+1,K), south (I,J-1,K), front (I,J,K+1), and back (I,J,K-1) or E, W, N, S, F, B respectively. The boundaries are labeled by lower case letters e, w, n, s, f, and b. The cell boundaries coincide with the physical boundaries to make the application of boundary conditions easier. The previous geometry was rectangular and used a uniform grid [Ref. 31]. In the spherical/cylindrical geometry, the radial grid is no longer uniform. A control volume in the generalized orthogonal coordinate system is represented by

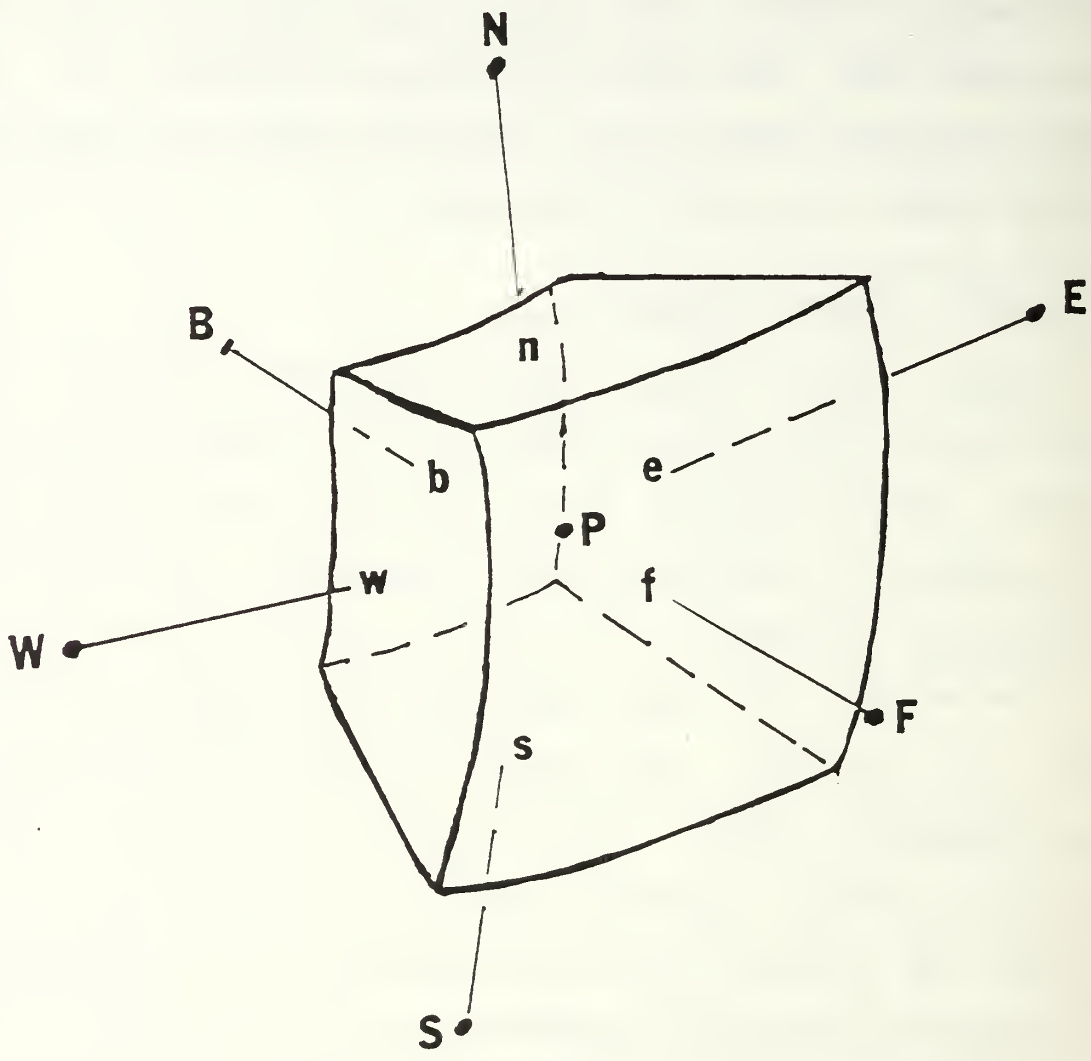


Figure 5.1 Spherical Basic Cell

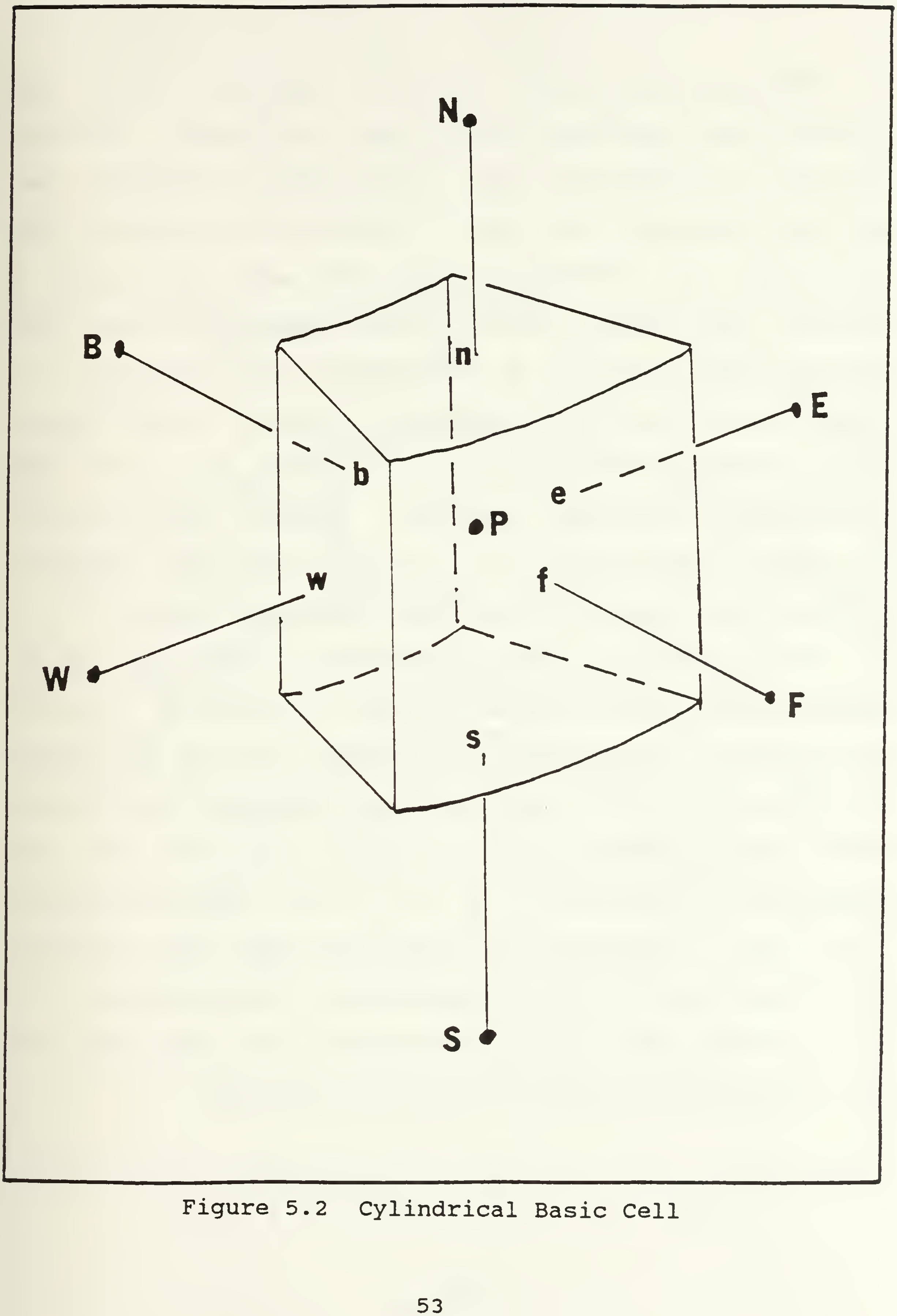


Figure 5.2 Cylindrical Basic Cell

$\sqrt{g}\,\theta^1\theta^2\,\theta^3$. The actual grid size will be discussed in Chapter VI.

Temperature,pressure, density, specific heat and viscosity are calculated at the basic grid point. Whereas velocity is calculated from a grid that is staggered one half cell from the basic grid. Difficulties can arise from calculating all variables from the same grid points. Patankar [Ref. 36:pp. 115-120] lists these difficulties and explains the advantage of a staggered grid. First, a staggered grid prevents a wavy, oscillatory, velocity field from satisfying the continuity equation by using the difference of adjacent velocities. Secondly, the velocity is easily calculated as a function of the pressure difference between the two adjacent basic grid points.

A two dimensional view to illustrate a basic cell and a staggered cell can be found in Figs. 5.3 and 5.4. The u^1 node labeled $\overline{P}$ corresponds to the west face of the basic cell centered at P. Its surrounding staggered cells have their centers marked as $\overline{E}$, $\overline{W}$, $\overline{N}$, $\overline{S}$, $\overline{F}$, $\overline{B}$ and the six boundaries are marked as $\overline{e}$, $\overline{w}$, $\overline{n}$, $\overline{s}$, $\overline{f}$, $\overline{b}$. The same applies to u^2, and u^3 components by moving the basic cell one-half cell to the south or back respectively. The velocity, u^1, for the basic cell (I,J,K) is located on its west face, u^2 is on the south face and u^3 is on the back face.

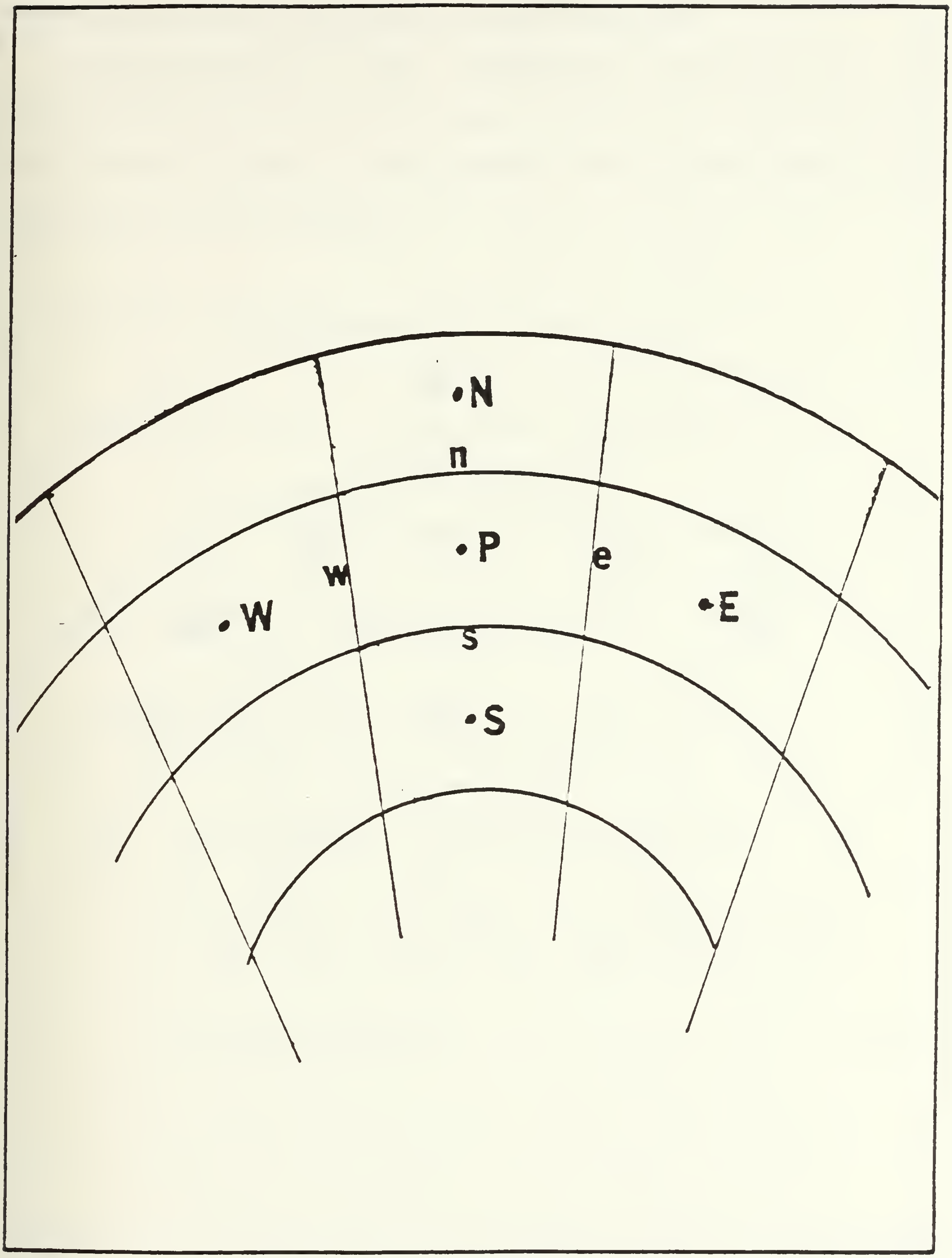


Figure 5.3 Two Dimensional Basic Cell

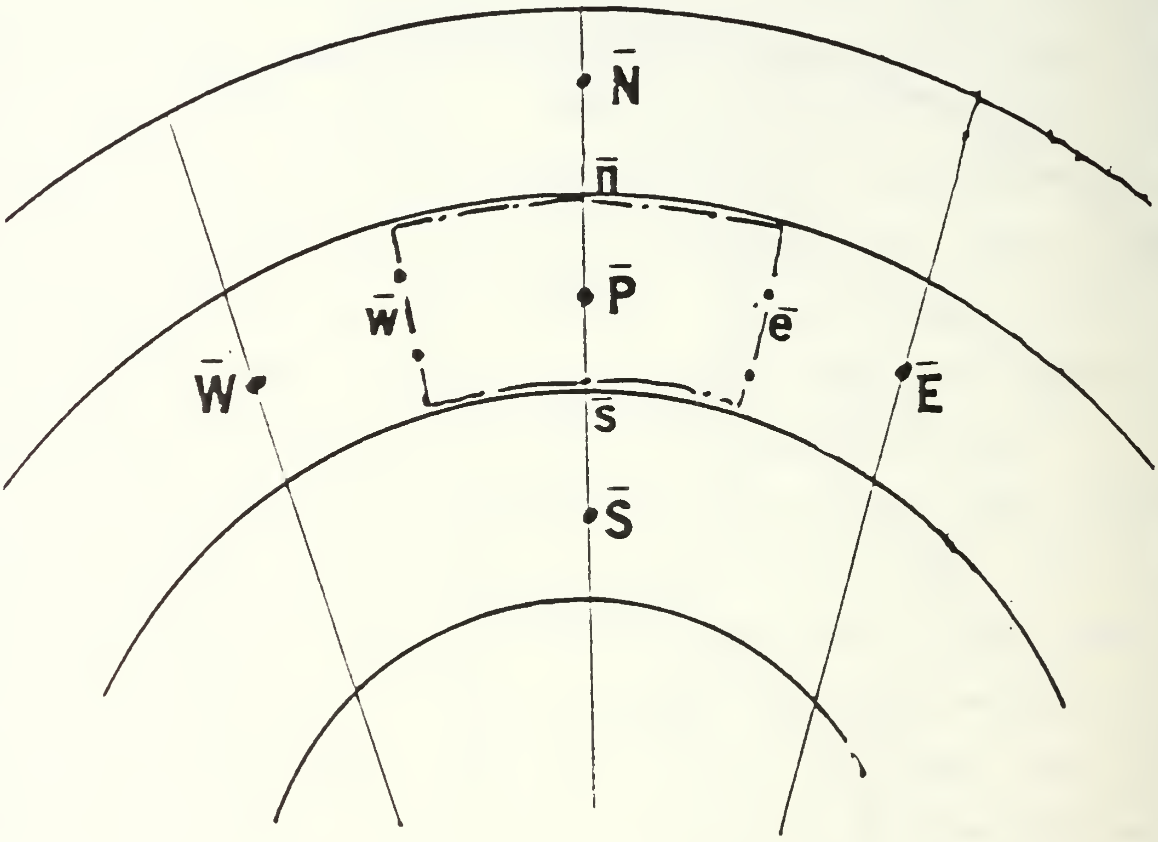


Figure 5.4 Two Dimensional Staggered Cell

C. INTEGRATION OF THE CONSERVATION EQUATIONS

Discretization of the conservation equations is now accomplished by integrating Eqns. 2.20, 2.21, and 2.23 over each control volume. The integral form of the conservation equations can be written:

Continuity Equation:

$$\int \frac{\partial \rho}{\partial t} h_1 h_2 h_3 d\theta^1 d\theta^2 d\theta^3 + \int \left[\frac{\partial}{\partial \theta^1}(\rho u^1 h_2 h_3) + \frac{\partial}{\partial \theta^2}(\rho u^2 h_3 h_1)\right.$$

$$\left. + \frac{\partial}{\partial \theta^3}(\rho u^3 h_1 h_2)\right] d\theta^1 d\theta^2 d\theta^3 = 0 \qquad (5.2)$$

Energy Equation:

$$\int \frac{\partial(\rho C_{pm} T)}{\partial t} h_1 h_2 h_3 d\theta^1 d\theta^2 d\theta^3 + \int \left[\frac{\partial}{\partial \theta^1}(\rho C_{pm} u^1 T h_2 h_3)\right.$$

$$\left. + \frac{\partial}{\partial \theta^2}(\rho C_{pm} u^2 T h_1 h_3) + \frac{\partial}{\partial \theta^3}(\rho C_{pm} u^3 T h_1 h_2)\right] d\theta^1 d\theta^2 d\theta^3$$

$$- \int \left[\frac{\partial}{\partial \theta^1}(q^1 h_2 h_3) + \frac{\partial}{\partial \theta^2}(q^2 h_1 h_3) + \frac{\partial}{\partial \theta^3}(q^3 h_1 h_2)\right] d\theta^1 d\theta^2 d\theta^3$$

$$+ \int S\, h_1 h_2 h_3 \, d\theta^1 d\theta^2 d\theta^3 \qquad (5.3)$$

where:

$$q^i = -\frac{k}{h_i}\frac{\partial T}{\partial \theta^i}$$

Momentum Equations:

$$\int \frac{\partial}{\partial t}(\rho u^i) h_1 h_2 h_3 \, d\theta^1 d\theta^2 d\theta^3 + \int \frac{\partial}{\partial \theta^j}\left[\left(\frac{h_1 h_2 h_3}{h_j}\right)\rho u^i u^j\right] d\theta^1 d\theta^2 d\theta^3$$

$$= \int -\frac{\partial}{\partial \theta^i}\left(P \frac{h_1 h_2 h_3}{h_i}\right) d\theta^1 d\theta^2 d\theta^3 + \int \rho G_i \, h_1 h_2 h_3 \, d\theta^1 d\theta^2 d\theta^3$$

$$+ \int \frac{\partial}{\partial \theta^j}\left(\sigma^{ij} \frac{h_1 h_2 h_3}{h_j}\right) d\theta^1 d\theta^2 d\theta^3$$

$$- \int \frac{h_1 h_2 h_3}{h_i h_j} \cdot \left[\frac{\partial h_i}{\partial \theta^j} (\rho u^j u^i - \sigma^{ij})\right] d\theta^1 d\theta^2 d\theta^3$$

$$+ \int \frac{h_1 h_2 h_3}{h_j h_i} \cdot \left[\frac{\partial h_j}{\partial \theta^i} (\rho u^j u^j - \sigma^{jj})\right] d\theta^1 d\theta^2 d\theta^3 \qquad (5.4)$$

For

u^1	Let	$i = 1$	$j = 1,2,3$
u^2	let	$i = 2$	$j = 1,2,3$
u^3	let	$i = 3$	$j = 1,2,3$

D. CONTINUITY EQUATION

In developing the finite differencing equations, finite quantities are substituted for the differential element in the integral form of the equations. Finite values are substituted for Δ quantities and for various fluxes across the cell boundaries. The differencing techniques used in this numerical scheme are forward differencing for the time

steps, central differencing for the diffusion terms, and QUICK for the convective terms.

In the forward differencing scheme, a future value of the dependent variable is predicted from the previous value plus a known slope multiplied by the time step. For continuity:

$$\rho^{n} = \rho^{n-1} + m\Delta t \tag{5.5}$$

where ρ^{n-1} is the old value for density at the previous time step, ρ is the new value for density, m is the known slope. The left hand side of Eqn. 5.2 becomes

$$\frac{\partial\rho}{\partial t}\, dV = \frac{\rho^{n} - \rho^{n-1}}{\Delta t}\, h_1 h_2 h_3\, \Delta\theta^1 \Delta\theta^2 \Delta\theta^3$$

$$= \frac{\rho^{n} - \rho^{n-1}}{\Delta t}\, \Delta V \tag{5.6}$$

Using the continuity equation as a model, the evaluation of the integral becomes

$$(\rho^{n} - \rho^{n-1})\, \frac{\Delta V}{\Delta t} + [\rho u^1 h_2 h_3 d\theta^2 d\theta^3]_e - [\rho u^1 h_2 h_3 d\theta^2 d\theta^3]_w$$

$$+ [\rho u^2 h_1 h_3 d\theta^1 d\theta^3]_n - [\rho u^2 h_1 h_3 d\theta^1 d\theta^3]_s$$

$$+ [\rho u^3 h_1 h_2 d\theta^1 d\theta^2]_f - [\rho u^3 h_1 h_2 d\theta^1 d\theta^2]_b = 0 \tag{5.7}$$

Because of the different locations for evaluating the density and velocity components, the symbol G will represent the mass flux rate. The mass flux rate will be evaluated at the six faces of the basic cell.

$$G_\rho = (\rho u^1)_e = u_e^1((\rho_p(h_1\Delta\theta^1)_{i+1} + \rho_E(h_1\Delta\theta^1)_i)/((h_1\Delta\theta^1)_{i+1} + (h_1\Delta\theta^1)_i)) \qquad (5.8)$$

$$G_w = (\rho u^1)_w = u_w^1((\rho_p(h_1\Delta\theta^1)_{i-1} + \rho_w(h_1\Delta\theta^1)_i)/((h_1\Delta\theta^1)_{i-1} + (h_1\Delta\theta^1)_i)) \qquad (5.9)$$

$$G_n = (\rho u^2)_n = u_n^2((\rho_p(h_2\Delta\theta^2)_{j+1} + \rho_N(h_2\Delta\theta^2)_j)/((h_2\Delta\theta^2)_{j+1} + (h_2\Delta\theta^2)_j)) \qquad (5.10)$$

$$G_s = (\rho u^2)_s = u_s^2((\rho_p(h_2\Delta\theta^2)_{j-1} + \rho_s(h_2\Delta\theta^2)_j)/((h_2\Delta\theta^2)_{j-1} + (h_2\Delta\theta^2)_j)) \qquad (5.11)$$

$$G_f = (\rho u^3)_f = u_f^3((\rho_p(h_3\Delta\theta^3)_{k+1} + \rho_f(h_3\Delta\theta^3)_k)/((h_3\Delta\theta^3)_{k+1} + (h_3\Delta\theta^3)_k)) \qquad (5.12)$$

$$G_b = (\rho u^3)_b = u_b^3((\rho_p(h_3\Delta\theta^3)_{k-1} + \rho_b(h_3\Delta\theta^3)_k)/((h_3\Delta\theta^3)_{k-1} + (h_3\Delta\theta^3)_k)) \qquad (5.13)$$

and the area is represented by

$$A_{e,w} = (h_2\,\Delta\theta^2 h_3\,\Delta\theta^3)_{e,w} \qquad (5.14)$$

$$A_{n,s} = (h_1\,\Delta\theta^1 h_3\,\Delta\theta^3)_{n,s} \qquad (5.15)$$

$$A_{f,b} = (h_1\,\Delta\theta^1 h_2\,\Delta\theta^2)_{f,b} \qquad (5.16)$$

The continuity equation is finite difference form is

$$\frac{(\rho^n - \rho^{n-1})\Delta V}{\Delta t} + G_e - G_w + G_n - G_s + G_f - G_b = S_{mp} \qquad (5.17)$$

where S_{mp} is the mass source term. If this were an ideal situation, the mass source term would be equal to zero. However, for an iterative numerical solution, the sum of the mass fluxes will equal a finite nonzero value, S_{mp}. As the solution is iterated and converges, the mass source term will approach zero. The solution will be iterated until S_{mp} is less than a predetermined cutoff value.

E. ENERGY EQUATION

The energy equation will be used to illustrate the QUICK scheme. Integration of the energy equation over the control volume leads to the following equation.

$$[(\rho C_{pm} T)^n - (\rho C_{pm} T)^{n-1}] \frac{\Delta V}{\Delta t} + G_e (C_{pm} T)_e A_e - G_w (C_{pm} T)_w A_w$$

$$+ G_n (C_{pm} T)_n A_n - G_s (C_{pm} T)_s A_s + G_f (C_{pm} T)_f A_f - G_b (C_{pm} T)_b A_b$$

$$= k_e A_e \left(\frac{\partial T}{h_1 \partial \theta^1}\right)_e - k_w A_w \left(\frac{\partial T}{h_1 \partial \theta^1}\right)_w + k_n A_n \left(\frac{\partial T}{h_2 \partial \theta^2}\right)_n$$

$$- k_s A_s \left(\frac{\partial T}{h_2 \partial \theta^2}\right)_s + k_f A_f \left(\frac{\partial T}{h_3 \partial \theta^3}\right)_f - k_b A_b \left(\frac{\partial T}{h_3 \partial \theta^3}\right)_b$$

$$+ S \, \Delta V \qquad (5.18)$$

where S is the source term which includes the terms of dissipation, pressure work, radiation and any internal heat sources (see Eqn. 2.26).

Let J represent the total heat flux which is due to convection and conduction.

$$J^1_{e,w} = [(\rho C_{pm} u^1 T) - k \frac{\partial T}{h_1 \partial \theta^1}]_{e,w} \tag{5.19}$$

$$J^2_{n,s} = [(\rho C_{pm} u^2 T) - k \frac{\partial T}{h_2 \partial \theta^2}]_{n,s} \tag{5.20}$$

$$J^3_{f,b} = [(\rho C_{pm} u^3 T) - k \frac{\partial T}{h_3 \partial \theta^3})_{f,b} \tag{5.21}$$

The above equations represent the θ^1, θ^2, θ^3 components of the total flux of heat. The subscript indicates the point to which they correspond. For example, J^1_e is the component of flux at point "e" on the east face; J^2_n is the component of flux at the point "n" on the north face; J^2_f is the component of flux at the point "f" on the front face. Substituting Eqns. 5.19 - 5.21 into Eqn. 5.18, the energy equation in finite difference form becomes:

$$[(\rho C_{pm} T)^n - (\rho C_{pm} T)^{n-1}] \frac{\Delta V}{\Delta t} + J^1_e A_e - J^1_w A_w$$

$$+ J^2_n A_n - J^2_s A_s + J^3_f A_f - J^3_b A_b = S \, \Delta V \tag{5.22}$$

In the heat expression, $(\rho u^1 C_{pm} T)$ gives rise to difficulties because C_{pm}, ρ, and T values are suppose to be evaluated at the center of the cell instead of the surface of the cell. The different flux components in Eqn. 5.22 must be expressed in terms of the value of C_{pm}, ρ, and T at the point P and its neighbors W, E, N, S, F, B.

1. QUICK Scheme

In deriving the finite difference equations, the main aim is to estimate accurate values of the dependent variables at the surfaces of the control volume with stable properties. One way to do this is by using the QUICK scheme. QUICK combines the stability of the upwind scheme with the relative accuracy of the central differencing scheme. This combination is achieved by using a parabolic polynomial interpolation to fit the control volume at three consecutive nodal positions. Two nodes located on either side of the surface and the third on the next node in the upstream direction. H.Q. Yang [Ref. 13:pp. 77-89] explains the use of QUICK for a one dimensional system and then expands to two and three dimensions. Before QUICK is applied to the generalized orthogonal system, a brief summary of H.Q. Yang's [Ref. 13:pp. 77-79] explanation of QUICK as it applies to a one dimensional cartesian coordinate system is repeated here.

The quadratic interpolation expression for a non-uniform grid spacing is given as (see Fig. 5.5):

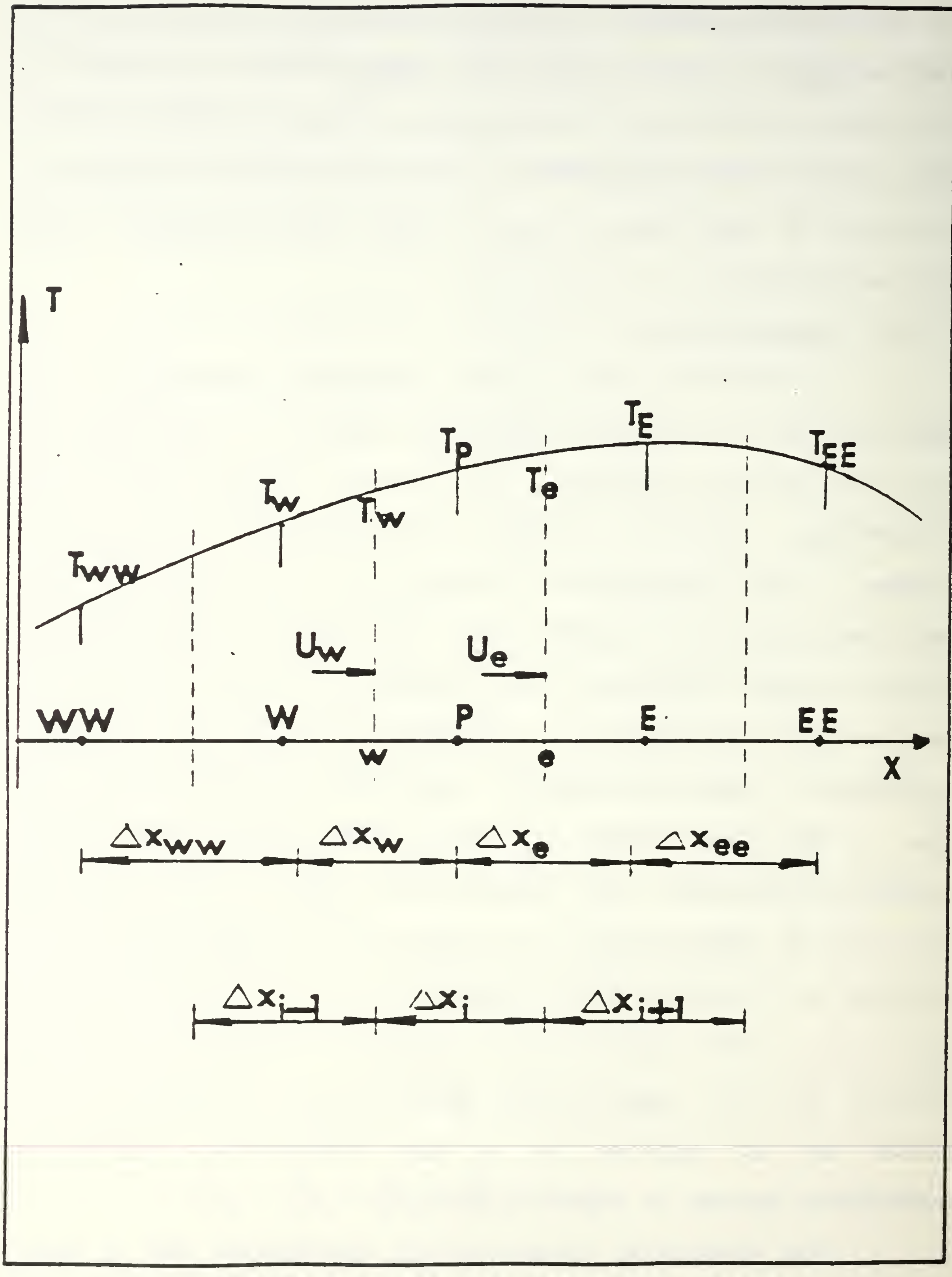


Figure 5.5 One Dimensional Quadratic Interpolation Scheme

$$(\rho C_{pm}uT)_e = G_eC_{pm.e}((T_p + T_E)/2 - 1/8\ CURV_e) \qquad (5.23)$$

$$(\rho C_{pm}VT)_w = G_wC_{pm.w}((T_p + T_w)/2 - 1/8\ CURV_w) \qquad (5.24)$$

where the upstream weighted curvature terms CURV are given by:

$$CURV_e = \Delta X_e^2/\Delta X_i((T_E - T_p)/\Delta X_e - (T_p - T_w)/\Delta X_w) \quad \text{if} \quad G_e > 0$$

$$= \Delta X_e^2/\Delta X_{i+1}((T_{EE} - T_E)/\Delta X_{ee} - (T_E - T_p)/\Delta X_e) \quad \text{if} \quad G_e < 0 \qquad (5.25)$$

$$CURV_w = \Delta X_w^2/\Delta X_{i+1}((T_p - T_W)/\Delta X_w - (T_W - T_{WW})/\Delta X_{ww} \quad \text{if} \quad G_w > 0$$

$$= \Delta X_w^2/\Delta X_i((T_E - T_P)/\Delta X_e - (T_p - T_W)/\Delta X_w) \quad \text{if} \quad G_W < 0 \qquad (5.26)$$

where

$$\begin{aligned} \Delta X_e &= .5(\Delta X_i + \Delta X_{i+1}) \\ \Delta X_w &= .5(\Delta X_i + \Delta X_{i-1}) \\ \Delta X_{ee} &= .5(\Delta X_{i+1} + \Delta X_{i+2}) \\ \Delta X_{ww} &= .5(\Delta X_{i-1} + \Delta X_{i-2}) \end{aligned} \qquad (5.27)$$

The expression modified for the generalized orthogonal coordinate system would be:

$$(\rho C_{pm}u^1T)_e = G_eC_{pm.e}((T_p + T_E)/2 - 1/8\ CURVN_e) \qquad (5.28)$$

$$(\rho C_{pm}u^2T)_w = G_w C_{pm.w}((T_p + T_w)/2 - 1/8\ CURVN_w) \qquad (5.29)$$

where

$$CURVN_e = (h_1\Delta\theta^1)_e^2/(h_1\Delta\theta^1)_i((T_E - T_p)/(h_1\Delta\theta^1)_e - (T_p - T_w)/(h_1\Delta\theta^1)_w) \quad \text{if } G_e > 0$$

$$= (h_1\Delta\theta^1)_e^2/(h_1\Delta\theta^1)_{i+1}((T_{EE}-T_E)/(h_1\Delta\theta^1)_{ee}-(T_E-T_p)/(h_1\Delta\theta^1)_e) \qquad (5.30)$$

$$\text{if } G_e < 0$$

Then,

$$CURVN_w = (h_1\Delta\theta^1)_w^2/(h_1\Delta\theta^1)_{i+1}((T_p-T_W)/(h_1\Delta\theta^1)_w-(T_W-T_{WW})/(h_1\Delta\theta^1)_{ww}) \quad \text{if } G_w > 0$$

$$= (h_1\Delta\theta^1)_w^2/(h_1\Delta\theta^1)_i((T_E-T_p)/(h_1\Delta\theta^1)_e-(T_p-T_W)/(h_1\Delta\theta^1)_w) \qquad (5.31)$$

$$\text{if } G_w < 0$$

$$(h_1\Delta\theta^1)_e = .5((h_1\Delta\theta^1)_i + (h_1\Delta\theta^1)_{i+1})$$

$$(h_1\Delta\theta^1)_w = .5((h_1\Delta\theta^1)_i + (h_1\Delta\theta^1)_{i-1})$$

$$(5.32)$$

$$(h_1\Delta\theta^1)_{ee} = .5((h_1\Delta\theta^1)_{i+1} + (h_1\Delta\theta^1)_{i+2})$$

$$(h_1\Delta\theta^1)_{ww} = .5((h_1\Delta\theta^1)_{i-1} + (h_1\Delta\theta^1)_{i-2})$$

The conventional finite difference form of Eqn. 5.22 for a one dimensional system is written:

$$(\rho C_{pm.p}T_p)_{,t}h_1\Delta\theta^1 = A_E T_E + A_W T_W - A_p T_p + S(h_1\Delta\theta^1) \qquad (5.33)$$

Using a semi-implicit tridiagonal solution procedure, T_{EE} and T_{WW} are incorporated into the source term. The other coefficients will be equal to: (for a uniform grid)

$$A_E = C_{pm.e}(-7G_e+3|G_e|)/16 + C_{pm.w}(-G_w+|G_w|)+ke/h_1\Delta\theta^1 \quad (5.34)$$

$$A_W = C_{pm.w}(9G_w+3|G_w|)/16 + C_{pm.e}(G_e+|G_e|)+k_w/h_1\Delta\theta^1 \quad (5.35)$$

$$A_p = 9(G_wC_{pm.w}-G_eC_{pm.e})/16 + 3(|G_w|C_{pm.w} + |G_e|C_{pm.e})/16$$

$$+ (k_w + k_e)/(h_1\Delta\theta^1) \quad (5.36)$$

$$S_p = Sh_1\Delta\theta^1 - C_{pm.e}(|G_e|-G_e)T_{EE} - C_{pm.w}(|G_w|+G_w)T_{WW} \quad (5.37)$$

The extension of the QUICK scheme to two and three dimensions is preformed by H.Q. Yang [Ref. 13:pp. 82-89]. Only the three dimensional algorithm as it applies to the generalized orthogonal coordinate system will be described here.

The 3-D QUICK algorithm is based on locally quadratic interpolation of temperature on each control volume. The average control volume temperature is found in a similar manner as the one dimensional case, only now there are more points to consider. A three dimensional representation of calculation cell with a uniform rectangular grid is found in Fig. 5.6. A similar situation applies to this geometry only a more complex figure would

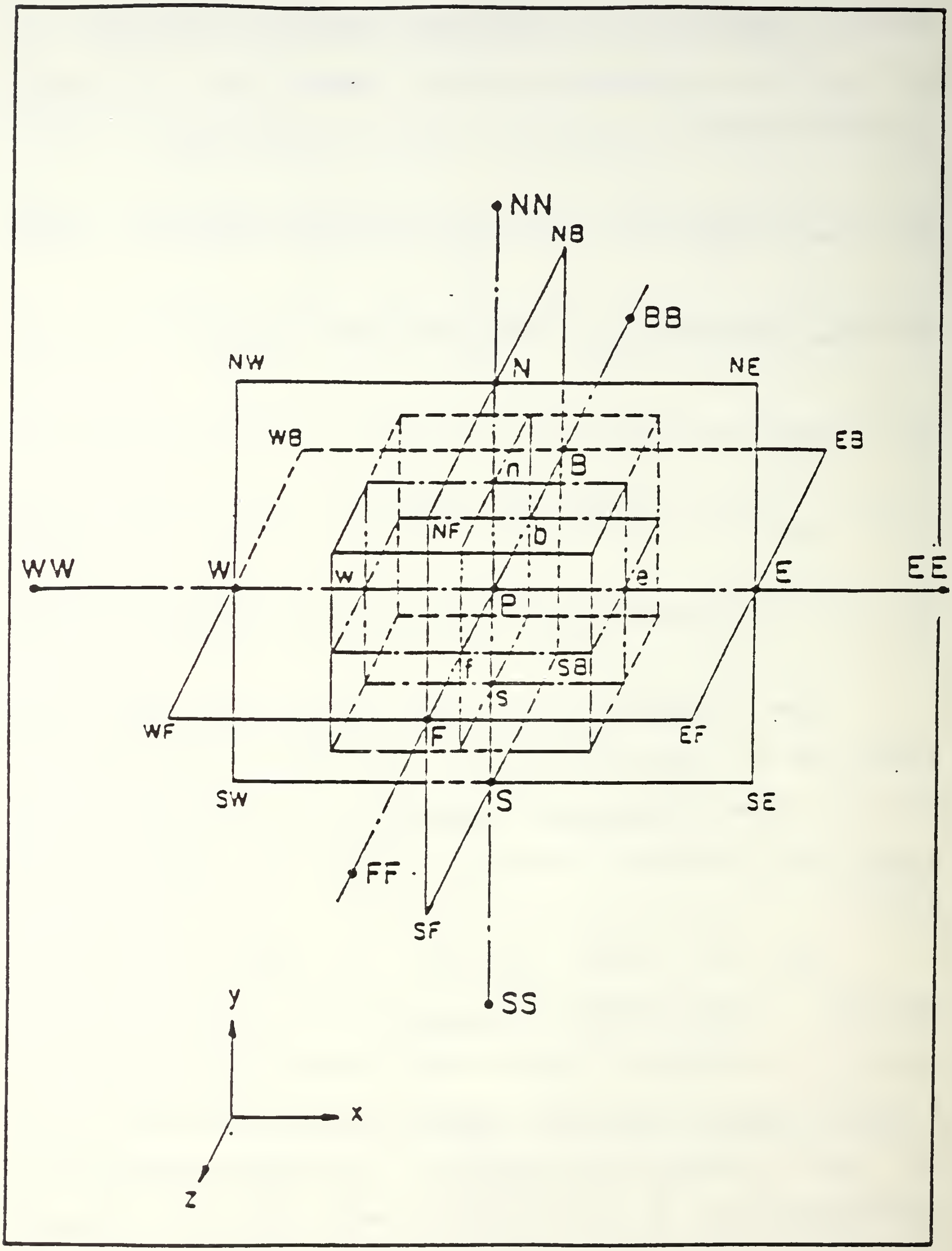


Figure 5.6 Calculation Cell for a Uniform Rectangular Grid

represent the calculation cell. The interested reader should refer to Yang [Ref. 13] for the evaluation of the temperature and curvilinear terms. The curvature terms are derived for each of the temperatures and substituted into the convection heat flux expressions. The heat flux is then found going into each surface of the control volume. Once the heat flux is found, it is substituted into Eqn. 5.22. After separating variables, the energy equation can be written as,

$$(A_p^T + (\rho C_{pm.p})^{n-1})\frac{\Delta V}{\Delta t} T_p = A_E^T T_E + A_W^T T_W + A_N^T T_N + A_s^T T_s$$

$$+ A_F^T T_F + A_B^T T_B + S_u^T \qquad (5.38)$$

where the additional source term is,

$$S_u^T = (\rho C_{pm.p} T)^{n-1} \frac{\Delta V}{\Delta t} - A_{EER} + A_{WWR} + A_{NNR} + A_{SSR}$$

$$+ A_{FFR} + A_{BBR} \qquad (5.39)$$

In the derivations that follow, all properties are assumed to be at (i,j,k) unless alternative values are given. If only one indice changes, that will be the only one so noted. For example the point (i+1,j,k) will only be denoted by i+1. If j and k are not given, they are assumed to remain unchanged.

$$CN = G_n * u^2_{j+1} * (h_3 \Delta\theta^3)_n (h_1 \Delta\theta^1)_n$$

$$CS = G_s * u^2_j * (h_3 \Delta\theta^3)_s (h_1 \Delta\theta^1)_s$$

$$CE = G_e * u^1_{i+1} * (h_2 \Delta\theta^2)_e) (h_3 \Delta\theta^3)_e$$

$$CW = G_w * u^1_i * (h_2 \Delta\theta^2)_w (h_3 \Delta\theta^3)_w \qquad (5.40)$$

$$CF = G_f * u^3_{k+1} * (h_1 \Delta\theta^1)_f (h_2 \Delta\theta^2)_f$$

$$CB = G_b * u^3_k * (h_1 \Delta\theta^1)_b) (h_2 \Delta\theta^2)_b$$

The thermal conductivity can be expressed as:

$$k_n = 1/((\frac{1}{k_j * (h_2\Delta\theta^2)_j} + \frac{1}{k_{j+1} * (h_2\Delta\theta^2)_{j+1}})/((h_2\Delta\theta^2)_j + (h_2\Delta\theta^2)_{j+1}))$$

$$k_s = 1/((\frac{1}{k_j * (h_2\Delta\theta^2)_j} + \frac{1}{k_{j-1} (h_2\Delta\theta^2)_{j-1}})/((h_2\Delta\theta^2)_j + (h_2\Delta\theta^2)_{j-1}))$$

$$k_e = 1/((\frac{1}{k_i * (h_1\Delta\theta^1)_i} + \frac{1}{k_{i+1} (h_1\Delta\theta^1)_{i+1}})/((h_1\Delta\theta^1)_i + (h_1\Delta\theta^1)_{i+1})) \qquad (5.41)$$

$$k_w = 1/((\frac{1}{k_i * (h_1\Delta\theta^1)_i} + \frac{1}{k_{i-1} (h_1\Delta\theta^1)_{i-1}})/((h_1\Delta\theta^1)_i + (h_1\Delta\theta^1)_{i-1}))$$

$$k_f = 1/((\frac{1}{k_k * (h_3\Delta\theta^3)_k} + \frac{1}{k_{k+1} (h_3\Delta\theta^3)_{k+1}})/((h_3\Delta\theta^3)_k + (h_3\Delta\theta^3)_{k+1}))$$

$$k_b = 1/((\frac{1}{k_k * (h_3\Delta\theta^3)_k} + \frac{1}{k_{k-1} (h_3\Delta\theta^3)_{k-1}})/((h_3\Delta\theta^3)_k + (h_3\Delta\theta^3)_{k-1}))$$

$$\mathrm{CONDN1} = k_n * [(h_3\Delta\theta^3 * h_1\Delta\theta^1)/h_2\Delta\theta^2]_n$$

$$\mathrm{CONDS1} = k_s * [(h_3\Delta\theta^3 * h_1\Delta\theta^1)/h_2\Delta\theta^2]_s$$

$$\mathrm{CONDE1} = k_e * [(h_2\Delta\theta^2 * h_3\Delta\theta^3)/h_1\Delta\theta^1]_e$$

$$\mathrm{CONDW1} = k_w * [(h_2\Delta\theta^2 * h_3\Delta\theta^3)/h_1\Delta\theta^1]_w \tag{5.42}$$

$$\mathrm{CONDF1} = k_f * [(h_1\Delta\theta^1 * h_2\Delta\theta^2)/h_3\Delta\theta^3]_f$$

$$\mathrm{CONDB1} = k_b * [(h_1\Delta\theta^1 * h_2\Delta\theta^2)/h_3\Delta\theta^3]_b$$

$$\mathrm{CEP} = (|\mathrm{CE}| + \mathrm{CE})((h_1\Delta\theta^1)_e/(h_1\Delta\theta^1)_i)\ \frac{1}{16}$$

$$\mathrm{CEM} = (|\mathrm{CE}| - \mathrm{CE})((h_1\Delta\theta^1)_e/(h_1\Delta\theta^1)_{i+1})\ \frac{1}{16}$$

$$\mathrm{CWP} = (|\mathrm{CW}| + \mathrm{CW})((h_1\Delta\theta^1)_w/(h_1\Delta\theta)_{i-1})\ \frac{1}{16}$$

$$\mathrm{CWM} = (|\mathrm{CW}| - \mathrm{CW})((h_1\Delta\theta^1)_w/(h_1\Delta\theta)_i)\ \frac{1}{16}$$

$$\mathrm{CNP} = (|\mathrm{CN}| + \mathrm{CN})((h_2\Delta\theta^2)_n/(h_2\Delta\theta^2)_j)\ \frac{1}{16}$$

$$\mathrm{CNM} = (|\mathrm{CN}| - \mathrm{CN})((h_2\Delta\theta^2)_n/(h_2\Delta\theta^2)_{j+1})\ \frac{1}{16} \tag{5.43}$$

$$\mathrm{CSP} = (|\mathrm{CS}| + \mathrm{CS})((h_2\Delta\theta^2)_s/(h_2\Delta\theta^2)_{j-1})\ \frac{1}{16}$$

$$\mathrm{CSM} = (|\mathrm{CS}| - \mathrm{CS})((h_2\Delta\theta^2)_s/(h_2\Delta\theta^2)_j)\ \frac{1}{16}$$

$$\mathrm{CFP} = (|\mathrm{CF}| + \mathrm{CF})((h_3\Delta\theta^3)_f/(h_3\Delta\theta^3)_k)\ \frac{1}{16}$$

$$\mathrm{CFM} = (|\mathrm{CF}| - \mathrm{CF})((h_3\Delta\theta^3)_f/(h_3\Delta\theta^3)_{k+1})\ \frac{1}{16}$$

$$\mathrm{CBP} = (|\mathrm{CB}| + \mathrm{CB})((h_3\Delta\theta^3)_b/(h_3\Delta\theta^3)_{k-1})\ \frac{1}{16}$$

$$\mathrm{CBM} = (|\mathrm{CB}| - \mathrm{CB})((h_3\Delta\theta^3)_b/(h_3\Delta\theta^3)_k)\ \frac{1}{16}$$

$$A^T_{EE} = -CEM * (h_1 \Delta\theta^1)_e/(h_1 \Delta\theta^1)_{ee}$$

$$A^T_{WW} = -CWP * (h_1 \Delta\theta^1)_w/(h_1 \Delta\theta^1)_{ww}$$

$$A^T_{NN} = -CNM * (h_2 \Delta\theta^2)_n/(h_2 \Delta\theta^2)_{nn}$$

$$A^T_{SS} = -CSP * (h_2 \Delta\theta^2)_s/(h_2 \Delta\theta^2)_{ss} \qquad (5.44)$$

$$A^T_{FF} = -CFM * (h_3 \Delta\theta^3)_f/(h_3 \Delta\theta^3)_{ff}$$

$$A^T_{BB} = -CBP * (h_3 \Delta\theta^3)_b/(h_3 \Delta\theta^3)_{bb}$$

$$A_{EER} = A^T_{EE} * T_{i+2} * C_{pm_{i+2}}$$

$$A_{WWR} = A^T_{WW} * T_{i-2} * C_{pm_{i-2}}$$

$$A_{NNR} = A^T_{NN} * T_{j+2} * C_{pm_{j+2}}$$

$$A_{SSR} = A^T_{SS} * T_{j-2} * C_{pm_{j-2}} \qquad (5.45)$$

$$A_{FFR} = A^T_{FF} * T_{k+2} * C_{pm_{k+2}}$$

$$A_{BBR} = A^T_{BB} * T_{k-2} * C_{pm_{k-2}}$$

The intermediate coefficients are:

$$A_{EI} = [-.5*CE + CEP + CEM*(1 + (h_1\Delta\theta^1)_e/(h_1 \Delta\theta^1)_{ee}) + CWM * ((h_1 \Delta\theta^1)_w/(h_1 \Delta\theta^1)_e)] \qquad (5.46)$$

$$A_{WI} = [.5*CW + CWM + CWP*(1 + (h_1 \Delta\theta^1)_w/(h_1 \Delta\theta^1)_{ww}) + CEP * ((h_1\Delta\theta^1)_e/(h_1 \Delta\theta^1)_w)] \quad (5.47)$$

$$A_{NI} = [-.5*CN + CNP + CNM*(1 + (h_2 \Delta\theta^2)_n/(h_2 \Delta\theta^2)_{nn}) + CSM * ((h_2 \Delta\theta^2)_s/(h_2\Delta\theta^2)_n)] \quad (5.48)$$

$$A_{SI} = [.5*CS + CSM + CSP*(1 + (h_2 \Delta\theta^2)_s/(h_2 \Delta\theta^2)_{ss}) + CNP * ((h_2\Delta\theta^2)_n/(h_2\Delta\theta^2)_s)] \quad (5.49)$$

$$A_{FI} = [-.5*CF + CFP + CFM*(1 + (h_3 \Delta\theta^3)_f/(h_3 \Delta\theta^3)_{ff}) + CBM * ((h_3\Delta\theta^3)_b/(h_3\Delta\theta^3)_f)] \quad (5.50)$$

$$A_{BI} = [.5*CB + CBM + CBP*(1 + (h_3 \Delta\theta^3)_b/(h_3 \Delta\theta^3)_{bb}) + CFP * ((h_3\Delta\theta^3)_f/(h_3\Delta\theta^3)_b)] \quad (5.51)$$

The coefficients are:

$$\begin{aligned}
A_E^T &= A_{EI} * C_{pm.E} + CONDE1 \\
A_W^T &= A_{WI} * C_{pm.W} + CONDW1 \\
A_N^T &= A_{NI} * C_{pm.N} + CONDN1 \\
A_S^T &= A_{SI} * C_{pm.s} + CONDS1 \\
A_F^T &= A_{FI} * C_{pm.F} + CONDF1 \\
A_B^T &= A_{BI} + C_{pm.B} + CONDB1
\end{aligned} \quad (5.52)$$

A_P^T is the summation of all the A's.

$$A_P^T = (A_E^T + A_W^T + A_N^T + A_S^T + A_F^T + A_B^T + A_{EE}^T + A_{WW}^T$$
$$+ A_{NN}^T + A_{SS}^T + A_{FF}^T + A_{BB}^T) * C_{pm.p}$$
$$+ CONDE1 + CONDW1 + CONDN2 + CONDS1$$
$$+ CONDF1 + CONDB1 \qquad (5.53)$$

F MOMENTUM EQUATION

Integration of the momentum equation over the control volume leads to the following equation [Ref. 20]

$$(\rho u^i)_t V + M_e^{i1} A_e - M_w^{i1} A_w + M_n^{i2} A_n - M_s^{i2} A_s + M_f^{i3} A_f$$
$$- M_b^{i3} A_b = S^i \qquad (5.54)$$

where if i = 1 , the momentum equation is for u^1, i = 2 the equation is for u^2, and if i = 3 the equation is for u^3. $A_{e,w}$, $A_{n,s}$, and $A_{f,b}$ are given in Eqns. 5.14-5.16 and represent areas on the staggered cell. M^{ij} is the total momentum flux along the θ^{ij} direction for the velocity component u^i due to convection and diffusion. The subscript for M in Eqn. 5.54 denotes the position where it is evaluated.

$$M^{ij} = (\rho u^i u^j - \sigma_i^j) \qquad (5.55)$$

The source term S includes the pressure gradient, body force, Coriolis force and centrifugal forces. The source term for u^1 is:

$$S^1 = -P_e A_e + P_w A_w + \rho G^1 \Delta V - M_p^{12}(A_n - A_s)$$

$$- M_p^{13}(A_f - A_b) + (M_p^{22} + M_p^{33})(A_e - A_w) \qquad (5.56)$$

The "stress-flux formulation" is used. The stresses are evaluated from prior information and the source is known at the present iteration. Yang et al. [Ref. 20:pp. 11-13] use the idea of "stress-flux formulation" as it applies to the curvilinear coordinate system. The momentum flux is given as:

$$M^{ij} = \hat{M}^{ij} + (\hat{\sigma}_i^j - \sigma_i^j) \qquad (5.57)$$

where

$$\hat{\sigma}_i^j = \mu/[h_j(\frac{\partial u^i}{\partial \theta^j})] \qquad (5.58)$$

$$\hat{M}^{ij} = \rho u^i u^j - \hat{\sigma}_i^j \qquad (5.59)$$

The momentum equation for u^1 is now expressed as,

$$(\rho u)_t + \hat{M}^{11}_e A_e - \hat{M}^{11}_w A_w + \hat{M}^{12}_n A_n - \hat{M}^{12}_s A_s$$

$$+ \hat{M}^{13}_f A_f - \hat{M}^{13}_b A_b = \hat{S} \qquad (5.60)$$

$$\hat{S} = S - (\hat{\sigma}^1_1 - \sigma^1_1)_e A_e + (\hat{\sigma}^1_1 - \sigma^1_1)_w A_w$$

$$- (\hat{\sigma}^2_1 - \sigma^2_1)_n A_n + (\hat{\sigma}^2_1 - \sigma^2_1)_s A_s$$

$$- (\hat{\sigma}^3_1 - \sigma^3_1)_f + (\hat{\sigma}^3_1 - \sigma^3_1)_b A_b \qquad (5.61)$$

The momentum equations are more complex since they are developed around a staggered cell. The additional sheer stress tensor also adds to the complexity.

The θ^i momentum equation takes almost the same form as the energy equation,

$$(A^{u^1}_p + \rho^{n-1} \Delta V/\Delta t) u^1_p = A^{u^1}_E u^1_E + A^{u^1}_W u^1_W$$

$$+ A^{u^1}_N u^1_N + A^{u^1}_s u^1_s + A^{u^1}_F u^1_F + A^{u^1}_B u^1_B + S^{u^1} u^1 \qquad (5.62)$$

Introducing intermediate mass flow rate per unit area:

$$G_{ne} = [(\rho_{j+1}(h_2\Delta\theta^2)_j+\rho_j(h_2\Delta\theta^2)_{j+1})/(h_2\Delta\theta^2)_j+(h_2\Delta\theta^2)_{j+1})]u^2_{j+1}$$

$$G_{nw} = [(\rho_{i-1,j+1}(h_2\Delta\theta^2)_j+\rho_{i-1}(h_2\Delta\theta^2)_{j+1})/((h_2\Delta\theta^2)_j+(h_2\Delta\theta^2)_{j+1}]u^2_{i-1,j+1}$$

$$G_{se} = [(\rho_{j-1}(h_2\Delta\theta^2)_j+\rho_j(h_2\Delta\theta^2)_{j-1})/((h_2\Delta\theta^2)_j+(h_2\Delta\theta^2)_{j+1})]u^2$$

$$G_{sw} = [(\rho_{i-1,j-1}(h_2\Delta\theta^2)_j+\rho_{i-1}(h_2\Delta\theta^2)_{j-1})/((h_2\Delta\theta^2)_j+(h_2\Delta\theta^2)_{j-1})]u^2_{i-1}$$

$$G_e = [(\rho_{i+1}(h_1\Delta\theta^1)_e+\rho_i(h_1\Delta\theta^1)_{ee})/((h_1\Delta\theta^1)_e+(h_1\Delta\theta^1)_{ee})]u^1_{i+1}$$

$$G_p = [(\rho_{i-1}(h_1\Delta\theta^1)_e+\rho_i(h_1\Delta\theta^1)_w)/((h_1\Delta\theta^1)_e+(h_1\Delta\theta^1)_w)]u^1 \tag{5.63}$$

$$G_w = [(\rho_{i-2}(h_1\Delta\theta^1)_w+\rho_{i-1}(h_1\Delta\theta^1)_{ww})/((h_1\Delta\theta^1)_w+(h_1\Delta\theta^1)_{ww})]u^1_{i-1}$$

$$G_{fe} = [(\rho_{k+1}(h_3\Delta\theta^3)_k+\rho_k(h_3\Delta\theta^3)_{k+1})/((h_3\Delta\theta^3)_k+(h_3\Delta\theta^3)_{k+1})]u^3_{k+1}$$

$$G_{fw} = [(\rho_{i-1,k+1}(h_3\Delta\theta^3)_k+\rho_{i-1}(h_3\Delta\theta^3)_{k+1})/((h_3\Delta\theta^3)_k+(h_3\Delta\theta^3)_{k+1})]u^3_{i-1,k+1}$$

$$G_{be} = [(\rho_{k-1}(h_3\Delta\theta^3)_k+\rho(h_3\Delta\theta^3)_{k-1})/((h_3\Delta\theta^3)_k+(h_3\Delta\theta^3)_{k-1})]u^3$$

$$G_{bw} = [(\rho_{i-1,k-1}(h_3\Delta\theta^3)_k+\rho_{i-1}(h_3\Delta\theta^3)_{k-1})/((h_3\Delta\theta^3)_k+(h_3\Delta\theta^3)_{k-1})]u^3_{i-1}$$

Final mass flow rates through each control volume surface are:

$$CE = .5(G_e + G_p) * (h_2\Delta\theta^2)_e * (h_3\Delta\theta^3)_e$$

$$CW = .5(G_p + G_w) * (h_2\Delta\theta^2)_w * (h_3\Delta\theta^3)_w$$

$$CN = [(G_{ne}*(h_1\Delta\theta^1)_w + G_{nw}(h_1\Delta\theta^1)_e)/((h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e)](h_1\Delta\theta^1)_n(h_3\Delta\theta^3)_n$$

(5.64)

$$CS = [(G_{se}(h_1\Delta\theta^1)_w + G_{sw}(h_1\Delta\theta^1)_e)/((h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e)](h_1\Delta\theta^1)_s(h_3\Delta\theta^3)_s$$

$$CF = [(G_{fe}(h_1\Delta\theta^1)_w + G_{fw}(h_1\Delta\theta^1)_e)/((h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e)](h_2\Delta\theta^2)_f(h_1\Delta\theta^1)_f$$

$$CB = [(G_{be}(h_1\Delta\theta^1)_w + G_{bw}(h_1\Delta\theta^1)_e)/((h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e)](h_2\Delta\theta^2)_b(h_1\Delta\theta^1)_b$$

The local viscosity is:

$$VIS_e = VIS$$

$$VIS_w = VIS_{i-1}$$

$$VIS_n = (VIS_{j+1} + VIS + VIS_{i-1,j+1} + VIS_{i-1})/4.0$$

(5.65)

$$VIS_s = (VIS_{j-1} + VIS + VIS_{i-1,j-1} + VIS_{i-1})/4.0$$

$$VIS_f = (VIS_{k+1} + VIS + VIS_{i-1,k+1} + VIS_{i-1})/4.0$$

$$VIS_b = (VIS_{k-1} + VIS + VIS_{i-1,k-1} + VIS_{i-1})/4.0$$

$$VISN1 = VIS_n * [(h_3\Delta\theta^3)(h_1\Delta\theta^1)/h_2\Delta\theta^2]_n$$

$$VISS1 = VIS_s * [(h_3\Delta\theta^3)(h_1\Delta\theta^1)/h_2\Delta\theta^2]_s$$

$$VISE1 = VIS_e * [(h_2\Delta\theta^2)(h_3\Delta\theta^3)/h_1\Delta\theta^1]_e \tag{5.66}$$

$$VISW1 = VIS_w * [(h_2\Delta\theta^2)(h_3\Delta\theta^3)/h_1\Delta\theta^1]_w$$

$$VISF1 = VIS_f * [(h_1\Delta\theta^1)(h_2\Delta\theta^2)/h_3\Delta\theta^3]_f$$

$$VISB1 = VIS_b * [(h_1\Delta\theta^1)(h_2\Delta\theta^2)/h_3\Delta\theta^3]_b$$

Equations that have the same form as those derived in the energy equation are also used here. These are Eqns. 5.43, 5.46-5.51, and 5.44. With the point other than the neighbor:

$$A_{EER} = A^u_{EE} * u^l_{i+2}$$

$$A_{WWR} = A^u_{WW} * u^l_{i-2}$$

$$A_{NNR} = A^u_{NN} * u^l_{j+2} \tag{5.67}$$

$$A_{SSR} = A^u_{SS} * u^l_{j-2}$$

$$A_{FFR} = A^u_{FF} * u^l_{k+2}$$

$$A_{BBR} = A^u_{BB} * u^l_{k-2}$$

all the coefficient A's are

$$A_E^u = A_{EI} + VISE1 \tag{5.68}$$

$$A_W^u = A_{WI} + VISWI \tag{5.69}$$

$$A_N^u = A_{NI} + VISNI \tag{5.70}$$

$$A_S^u = A_{SI} + VISSI \tag{5.71}$$

$$A_F^u = A_{FI} + VISFI \tag{5.72}$$

$$A_B^u = A_{BI} + VISB1 \tag{5.73}$$

and A_p^u is the summation of all the A's:

$$A_p^u = A_E^u + A_W^u + A_N^u + A_s^u + A_F^u + A_B^u + A_{EE}^u + A_{WW}^u$$

$$+ A_{NN}^u + A_{SS}^u + A_{FF}^u + A_{BB}^u \tag{5.74}$$

The source term is expressed as,

$$S_u^u = (\rho(h_1\Delta\theta^1)_w + \rho_{i-1}(h_1\Delta\theta^1)_e)/((h_1\Delta\theta^1)_e + (h_1\Delta\theta^1)_w)*\frac{\Delta V}{\Delta t}*u^1$$

$$+ (h_2\Delta\theta^2)_j(h_3\Delta\theta^3)_k[P_{i-1} - P_i] + A_{EER} + A_{WWR} + A_{NNR}$$

$$+ A_{SSR} + A_{FFR} + A_{BBR} + RE - RW + RN - RS + RF - RB$$

$$+ RRY + RRZ - RRX - Buoy * [\sin(ZC(k)*(\rho-\rho_{EQ})$$

$$*(h_1\Delta\theta^1)_w*\cos(XC(I))] + [(\rho_{i-1}-\rho_{EQ_{i-1}})(h_1\Delta\theta^1)_e$$

$$* \cos(XC(I-1))]/((h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e)\Delta V \qquad (5.75)$$

where XC and ZC represent the center of the basic cell. The rest of the variables in the source equation can be found in the following equations.

$$RE = (\sigma^{11} - (u^1_{i+1} - u^1_i)*VIS_e/(h_1\Delta\theta^1)_e)(h_2\Delta\theta^2\ h_3\Delta\theta^3)_e$$

$$RW = (\sigma^{11}_{i-1} - (u^1 - u^1_{i-1})*VIS_u/(h_1\Delta\theta^1)_w)(h_2\Delta\theta^2\ h_3\Delta\theta^3)_w$$

$$RN = (\sigma^{12}_{j+1} - (u^1_{j+1} - u^1_j)*VIS_n/(h_2\Delta\theta^2)_n)(h_1\Delta\theta^1\ h_3\Delta\theta^3)_n \qquad (5.76)$$

$$RS = (\sigma^{12} - (u^1 - u^1_{j-1})*VIS_s/(h_2\Delta\theta^2)_s)(h_1\Delta\theta^1\ h_3\Delta\theta^3)_s$$

$$RF = (\sigma^{13}_{k+1} - (u^1_{k+1} - u^1_k)*VIS_f/(h_3\Delta\theta^3)_f)(h_1\Delta\theta^1\ h_2\Delta\theta^2)_f$$

$$RB = (\sigma^{13} - (u^1 - u^1_{k-1})*VIS_b/(h_3\Delta\theta^3)_b)(h_1\Delta\theta^1\ h_2\Delta\theta^2)_b$$

$$\bar{\sigma}^{12} = .5(\sigma^{12}_{j+1} + \sigma^{12}_{j})$$

$$\bar{\sigma}^{13} = .5(\sigma^{13}_{k+1} + \sigma^{13}_{k})$$

$$\bar{\sigma}^{22} = (\sigma^{22})h_1\Delta\theta^1)_w + \sigma^{22}_{i-1}(h_1\Delta\theta^1)_e)/((h_1\Delta\theta^1)_w+(h_1\Delta\theta^1)_e) \tag{5.77}$$

$$\bar{\sigma}^{33} = (\sigma^{13}(h_1\Delta\theta^1)_w + \sigma^{33}_{i-1}(h_1\Delta\theta^1)_e)/((h_1\Delta\theta^1)_w+(h_1\Delta\theta^1)_e)$$

$$AU1 = u^1$$

$$AU2 = \{[(u^2_{j+1}(h_2\Delta\theta^2)_j+u^2_j(h_2\Delta\theta^2)_j)/2(h_2\Delta\theta^2)_j](h_1\Delta\theta^1)_w + [(u^2_{i-1,j+1}(h_2\Delta\theta^2)_j$$

$$+ u^2_{i-1}(h_2\Delta\theta^2)_j)/2(h_2\Delta\theta^2)_j](h_1\Delta\theta^1)_e\}/((h_1\Delta\theta^1)_e+(h_1\Delta\theta^1)_w)$$

$$AU3 = \{[(u^3_{k+1}(h_3\Delta\theta^3)_k+u^3(h_3\Delta\theta^3)_k)/2(h_3\Delta\theta^3)_k](h_1\Delta\theta^1)_w+[(u^3_{i-1,k+1}(h_3\Delta\theta^3)_k \tag{5.78}$$

$$+ u^3_{i-1}(h_3\Delta\theta^3)_k)/2(h_3\Delta\theta^3)_k](h_1\Delta\theta^1)_e\}/((h_1\Delta\theta^1)_e+(h_1\Delta\theta^1)_w)$$

$$AR = (\rho(h_1\Delta\theta^1)_w + \rho_{i-1}(h_1\Delta\theta^1)_e)/((h_1\Delta\theta^1)_w + (h_1\Delta\theta^1)_e)$$

$$ARU12 = AR * AU1 * AU2$$

$$ARU13 = AR * AU1 * AU3 \tag{5.79}$$

$$ARU22 = AR * AU2 * AU2$$

$$ARU33 = AR * AU3 * AU3$$

$$RRY = (\overline{\sigma}^{12}-ARU12) * (h_3\Delta\theta^3)_k * ((h_1\Delta\theta^1)_n-(h_1\Delta\theta^1)_s)$$

$$RRZ = (\overline{\sigma}^{13}-ARU13) * (h_2\Delta\theta^2)_j * ((h_1\Delta\theta^1)_f-(h_1\Delta\theta^1)_b) \qquad (5.80)$$

$$RRX = (\overline{\sigma}^{22}-ARU22) * (h_3\Delta\theta^3)_k * ((h_2\Delta\theta^2)_e-(h_2\Delta\theta^2)_w)$$

$$+ (\overline{\sigma}^{33}-ARU33) * (h_2\Delta\theta^2)_j * ((h_3\Delta\theta^3)_e-(h_3\Delta\theta^3)_w)$$

The momentum equations for θ^2 and θ^3 follow the same form, but are omitted for the sake of brevity.

G. PRESSURE CORRECTION

The finite difference equations for energy and momentum are used to solve for the temperature T and velocity components u^1, u^2, u^3,. The other two dependent variables, density ρ and pressure P, are related through the equation of state and the mass conservation Eqn. 5.17. As Doria [Ref. 37] pointed out, pressure is only weakly coupled to the equation of state. Therefore, the density is found from the equation of state by the use of updated temperatures and pressures. The mass conservation equation is used to correct for the pressure across each cell to ensure the mass is conserved.

The one disadvantage of using primitive variables is in the difficulty of calculating the pressure. In a closed system, the pressure changes everywhere if there is a net energy change in the system. If this happens a global pressure correction must be applied. A local pressure

correction is also used to account for changes in pressure within a region of the tank which determines the velocity field.

1. <u>Global Pressure Correction</u>

Nicolette, et al. [Ref. 4] developed a global pressure correction scheme for a two dimensional square enclosure. This scheme can be easily extended to the spherical/cylindrical geometry of Fire-1.

The overall pressure levels are increased or decreased in a system of constant mass and volume depending on whether energy is added or removed. Knowing the mass of the system can not change, the density at equilibrium times the cell volume is equal to a constant mass value. To find the total mass of the system, the mass of each cell is added. At any particular time step later, the mass must still equal the total mass of the system at equilibrium. Summing over N cells,

$$\sum \rho_i^n (\Delta V)_i = \sum \rho_{EQ,i} (\Delta V)_i \qquad (5.81)$$

where n indicates some time step later, and EQ indicates at equilibrium.

The density is a function of pressure and temperature only when the cell volume is constant and an ideal gas assumption is made. Using a * to indicate an estimated value and a ' with a subscript g as the global

correction, the true values of pressure and temperature can be expressed as,

$$P = P^* + P_g' \tag{5.82}$$

$$T = T^* + T_g' \tag{5.83}$$

where at any time step the true value is the sum of the estimated value and the global correction. Substituting Eqns. 5.82 and 5.83 into Eqn. 5.81 and applying the ideal gas law, the global pressure correction can now be found.

$$P_g' = \{\sum P_{EQ}[\frac{\Delta V}{T_i} - \frac{\Delta V}{T^*}] - \sum (P^*\Delta V/T^*)\}/\sum \frac{\Delta V}{T^*} \tag{5.84}$$

The global pressure correction is added to P*, which for the first guess is generally the pressure at the previous time step. This procedure is continued until a pressure is obtained to conserve mass in every cell.

2. Local Pressure Correction

A local pressure correction procedure is described by Patankar [Ref. 36:pp. 120-126] and Doria [Ref. 37:pp. 26-32]. First the pressure field is guessed at a given time step. This is usually the pressure field at the previous time step. The velocities are computed based on this pressure distribution. These velocities are used in the continuity equation. The residual mass source term, S_{mp}, is

calculated for each cell. A sum of the absolute values of S_{mp} gives an overall error for the conservation of mass for the system. If the residual mass source term does not fall below some predetermined value, the pressure must be corrected so that S_{mp} is reduced. Using the corrected pressure, new values for the velocities are found. The process can be repeated using the corrected pressure as the new guessed value. The final pressure obtained will be a result of satisfying the mass conservation equation within the desired accuracy. Once the pressures are known, the densities for the next time step are found using the equation of state.

The actual pressure equals a guess value plus a correction.

$$P = P^* + P' \tag{5.85}$$

where a ' indicates a local correction, * still indicates a guessed value.

The finite difference equation for the pressure correction is in a similar form to the other finite difference equations.

$$A_P P_P' = A_E P_E' + A_W P_W' + A_N P_N' + A_S P_S' + A_F P_F' + A_B P_B' - S_{mp} \Delta V \tag{5.86}$$

where

$$\rho_n = (\rho(h_2\Delta\theta^2)_{j+1} + \rho_{j+1}(h_2\Delta\theta^2))/((h_2\Delta\theta^2)_{j+1} + (h_2\Delta\theta^2))$$

$$\rho_s = (\rho(h_2\Delta\theta^2)_{j-1} + \rho_{j-1}(h_2\Delta\theta^2))/((h_2\Delta\theta^2)_{j-1} + (h_2\Delta\theta^2))$$

$$\rho_e = (\rho(h_1\Delta\theta^1)_{i+1} + \rho_{i+1}(h_1\Delta\theta^1))/((h_1\Delta\theta^1)_{i+1} + (h_1\Delta\theta^1)) \quad (5.87)$$

$$\rho_w = (\rho(h_1\Delta\theta^1)_{i-1} + \rho_{i-1}(h_1\Delta\theta^1))/((h_1\Delta\theta^1)_{i-1} + (h_1\Delta\theta^1))$$

$$\rho_f = (\rho(h_3\Delta\theta^3)_{k+1} + \rho_{k+1}(h_3\Delta\theta^3))/((h_3\Delta\theta^3)_{k+1} + (h_3\Delta\theta^3))$$

$$\rho_b = (\rho(h_3\Delta\theta^3)_{k-1} + \rho_{k-1}(h_3\Delta\theta^3))/((h_3\Delta\theta^3)_{k-1} + (h_3\Delta\theta^3))$$

$$A_E = \rho_e * (h_2\Delta\theta^2\ h_3\Delta\theta^3)^2_e/(A^{u^1}_{p_{i+1}} + \rho_e\ \Delta V/\Delta t) \quad (5.88)$$

$$A_W = \rho_w * (h_2\Delta\theta^2\ h_3\Delta\theta_3)^2_w/(A^{u^1}_p + \rho_w\ \Delta V/\Delta t) \quad (5.89)$$

$$A_N = \rho_n * (h_1\Delta\theta^1\ h_3\Delta\theta^3)^2_n/(A^{u^2}_{p_{j+1}} + \rho_n\ \Delta V/\Delta t) \quad (5.90)$$

$$A_S = \rho_s * (h_1\Delta\theta^1\ h_3\Delta\theta^3)^2_s/(A^{u^2}_p + \rho_s\ \Delta V/\Delta t) \quad (5.91)$$

$$A_F = \rho_f * (h_1\Delta\theta^1\ h_2\Delta\theta^2)^2_f/(A^{u^3}_p + \rho_f\ \Delta V/\Delta t) \quad (5.92)$$

$$A_B = \rho_b * (h_1\Delta\theta^1\ h_2\Delta\theta^2)^2_b/(A^{u^3}_p + \rho_b\ \Delta V/\Delta t) \quad (5.93)$$

$$A_P = A_E + A_W + A_N + A_S + A_F + A_B \quad (5.94)$$

Once the pressure correction is known, it is added to the guessed value to obtain the actual pressure. The new velocities are found from the following equations.

$$u^1 = u^{1*} + u^{1'} \tag{5.95}$$

$$u^2 = u^{2*} + u^{2'} \tag{5.96}$$

$$u^3 = u^{3*} + u^{3'} \tag{5.97}$$

where

$$u^{1'} = (P_p - P_w)(h_2 \Delta\theta^2 \, h_3 \quad {}^3)_p / (A_p^{u^1} + \rho_w \, \Delta V/\Delta t) \tag{5.98}$$

$$u^{2'} = (P_p - P_s)(h_1 \Delta\theta^1 \, h_3 \, \Delta\theta^3)_p / (A_p^{u^2} + \rho_s \, \Delta V/\Delta t) \tag{5.99}$$

$$u^{3'} = (P_p - P_b)(h_1 \Delta\theta^1 \, h_2 \, \Delta\theta^2)_p / (A_p^{u^3} + \rho_b \, \Delta V/\Delta t) \tag{5.100}$$

The value for S_{mp} is computed. If it does not fall within the desired limits, a new value for P' is computed. The cycle continues until S_{mp} satisfies the limits put upon it.

VI. COMPARISON OF THE NUMERICAL RESULTS WITH THE EXPERIMENTAL DATA

As noted before, the computer model developed here is designed to simulate fires in Fire-1, the test facility maintained at NRL. The general theory behind the development of the computer model has been given in the preceding chapters. This chapter will describe the computer model as it specifically applies to Fire-1. A brief explanation of the required parameters will be given, along with the solution procedure for the computer code. Three test cases will then be compared with the experimental results obtained in Fire-1.

The first test case (Trial 1), followed the procedure of Nies [Ref. 31]. The required heat release data was generated from the pressure curve. A pressure tracking routine was applied, forcing the calculated pressure to follow the experimental curve. In this case, only the temperatures can be compared to the experimental data to verify the computer code.

The second test case (Trial 2), used given heat release data. NRL did provide burn rate data that was known to be inaccurate. The inaccuracy was assumed to be the result of applying an incorrect scaling factor to the data. It seemed plausible to try and use the data to observe the general trend of the pressure and temperature. In this case,

pressure and temperature can both be used to verify the computer code.

The third test case (Trial 3), is a combination of the two preceeding test cases. The heat release curve is modified to follow the same trend as in Trial 2, but at a magnitude determined by Trial 1.

The computer program will generate pressure, temperature, and velocity fields. The overall pressure of the tank at each time step will be compared to the experimental results. The computer program will also calculate the temperature located at a position corresponding to a thermocouple in Fire-1 (see Fig 1.1). Experimental readings for three thermocouples will be compared to the computer model for each test case.

Besides a direct comparison to the experimental results, velocity and temperature fields from Trial 3 will be plotted at various time intervals and at selected cross sections. This will provide a way to show the development of recirculating flow patterns and penetration of heat with time.

A. NUMERICAL SIMULATION PARAMETERS

Specific parameters must be defined in order to represent a particular fire scenario. The following items must be known: initial conditions, type of fuel used and its burn rate, location of the fire, location of the thermocouples, physical dimensions and material composition

of the tank, plus any other items included in the tank, ie. decks, fans, etc. The computer code represents a fire in Fire-1. The physical description along with the sensor locations for Fire-1 can be found in Chapter I. Fire-1 is made of 3/8" ASTM--285 Grade C steel. The material properties required for the program are listed in Table 2.

TABLE 2

SPECIFIC PARAMETERS

WALL CHARACTERISTICS	
Thickness	3/8 in
Specific Heat	.1 Btu/(lbm F)
Thermal Conductivity	25 Btu/(hr ft F)
Density	487 lbm/ft^3
FIRE CHARACTERISTICS	
Type of Fuel	Methanol
Burn Rate	Provided
Initial Temperature	35.6^{o}C
Initial Pressure	1.0 ATM
Location	Center of Fire-1
	23.1 ft from endcap 3.21 ft from bottom

The fire scenario considered here does not include any decks, fans, or anything else in the interior of Fire-1 besides the fire itself. The fuel used was methanol. The

location of the fire was at the center of the tank; 23.1 feet from either endcap, and elevated 3.21 feet from the bottom of the tank. The ambient, or initial, temperature and pressure was 35.6°C and 1 atmosphere respectfully.

Once the physical characteristics are entered into the program, the grid size and time step must be determined. Due to the spherical/cylindrical geometry of Fire-1, a uniform grid is not practical. The grid is represented by Figures 6.1 and 6.2. Instead of an X,Y,Z, grid, it is a θ, R, ϕ grid for the spherical endcaps and θ, R, Z grid for the cylindrical midsection. The theta direction has 20 cells. The R direction has 12 cells representing the interior of the tank and one cell representing the tank wall. Another cell surrounds the vicinity of r = 0 and is used to avoid singularity. Each spherical endcap also has one cell for singularity problems along with a division of five cells. The cylindrical midsection has 18 cells in the Z direction. Further information can be found in Table 3. Increasing the number of grid points was not practical because of the large amount of CPU time already required for this grid.

The stability of the computational results depended on the time step chosen. Initial calculations were done with a time step of .0288 sec. Once instability was reached, the program was continued with a smaller time step. The bulk of the program was accomplished at a time step of .0192 sec. Approximately 80 time steps were accomplished per hour of

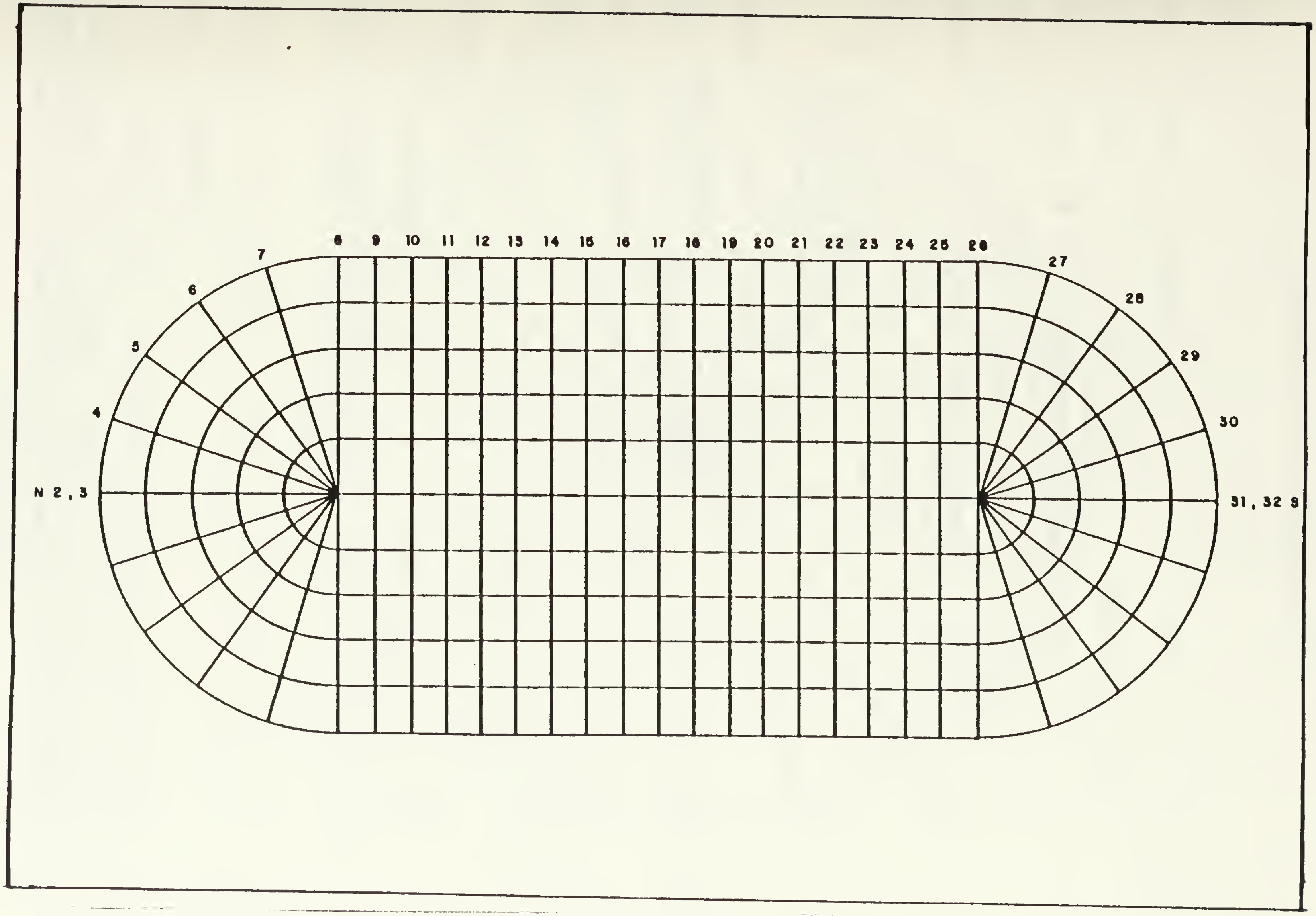


Figure 6.1 Vertical Cut Through the Computer Model

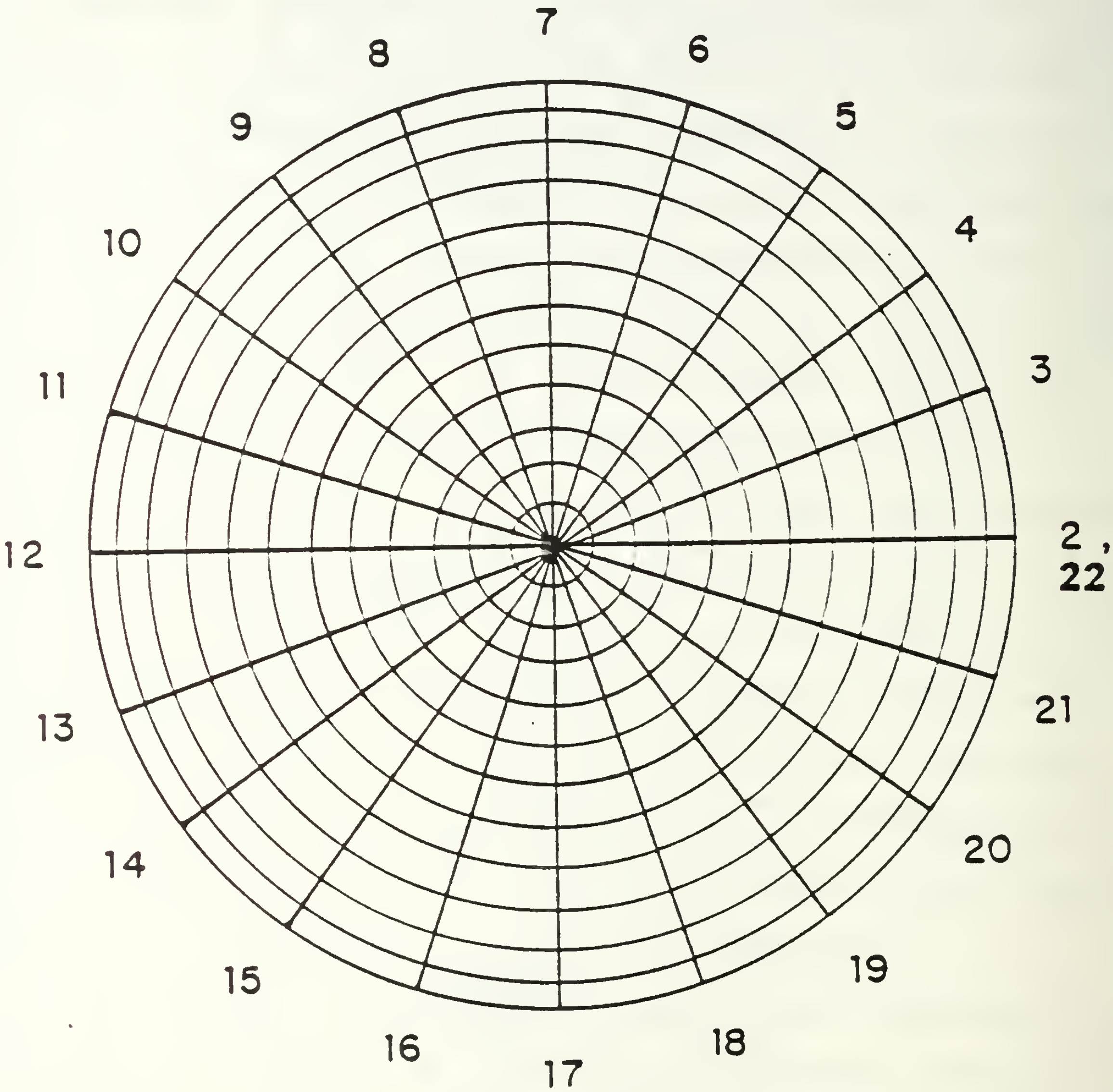


Figure 6.2 Cylindrical Cross Section of the Computer Model

TABLE 3

COMPUTER MODEL PARAMETERS

GRID	
Total number of interior cells	6720 (20 x 12 x 28)
Total number of tank wall cells	560 (20 x 28)
Total number of wall radiation zones	560
Total number of fire radiation zones	19
Number of cells in the theta direction	20
Number of cells in the R direction	14
- interior cells	12
- wall cells	1
- r = 0 , stability	1
Number of cells in the phi direction	12
- each end cap	5
- stability (each end cap)	1
Number of cells in the z direction	18
TIME STEP	
Varied	0.0192 - .0288 sec
CPU TIME	
80 time steps/CPU hr	~ 1 sec fire time
EXTERNAL HEAT TRANSFER COEFFICIENT	5.0 Btu/(hr ft F)

CPU time. This averaged to one sec of fire time for every hour of CPU time.

The external heat transfer coefficient is not known. It is estimated and becomes another parameter that can be changed.

B. SOLUTION PROCEDURE

The total computer model is actually composed of two separate programs. The radiation program in Appendix A calculates the view factors and then inverts them to provide the main program with a matrix of the form described in Eqn. (3.17). This program only needs to be run once with the results stored and read when needed.

The main program follows the same type of flow chart as described by Nies [Ref. 31:pp. 56-57]. After the initial parameters are read, the effective viscosity is found using the subroutine CALVIS. The radiation flux to the walls is updated every two time steps. The subroutines to calculate temperature, pressure, and velocity, all use a semi-implicit technique to solve the matrices formed by applying the finite difference equations discussed in Chapter V. First the temperature is found using CALT, then the global pressure is found followed by the density. An iteration loop is entered to find the three velocity components and local pressure correction. The local pressure correction is then used to update the velocities. The continuity equation is applied to each cell to calculate the residual mass. The sum of the absolute value of all the residual mass terms is called RESORM. If this term is extremely large, i.e., greater than ten, the program will stop indicating an instability problem. In the past this problem was resolved by lowering the time step. If the RESORM term is larger

than the tolerance value and below ten, the program will iterate the solution by recalculating the velocities and pressures. Because of the large amount of CPU time already required, the temperature, global pressure, and density are recalculated every third iteration. This also allows for the velocities to stabilize before changing the temperature. The iterative procedure will continue before proceeding to the next time step unless one of the following three things happen; the maximum number of iterations has been reached, the RESORM term is less than the tolerance, or if the CPU time for that particular run is almost exhausted.

C. TRIAL 1--PRESSURE TRACKING

Because of the uncertainty of the burn rate data provided by NRL, it was decided to use the method developed by Nies [Ref. 31] for one case. This method is a temporary solution pending accurate burn rate data. The first approximation is to assume the energy provided to a cavity is used almost exclusively to raise the pressure. This assumes the conduction through the tank walls is minimal and the motion of the gas does not require a large percentage of the energy.

Assuming the heat input is uniform, the rate of heat input is a constant times the temperature. Using the ideal gas law $P = \rho RT$ with ρ and R being constant, the heat release rate is proportional to the change in pressure with

respect to time. From the experimental pressure curve, the first derivative can be found.

Nies [Ref. 31] developed a pressure tracking routine to force the calculated pressure to follow the experimental pressure curve. This was done to account for the conduction losses as time increases.

The correction factor is computed as follows:

$$\text{Correction} = \frac{P_{data}-P_{comp}}{P_{data}} - \frac{P_{comp}-P^{o}_{comp}}{P_{data}} + 1 \qquad (6.1)$$

where P_{comp} is the computed pressure, P^{o}_{comp} is the computed pressure from the previous time step, and P_{data} is the experimental pressure.

1. <u>Numerical Results of Trial 1</u>

The computed heat input curve has oscillations present, Fig. 6.3. Nies [Ref. 31] had a similar result. Oscillations will enter into this case due to the correcting scheme and the use of the derivative of numerical data.

The pressure of the tank over 240 sec of fire time is found in Figure 6.4. This curve illustrates how the pressure tracking routine forced the calculated values of pressure to follow the curve fit through the experimental points.

The only other way to verify the model is to compare the thermocouple temperatures with the temperatures obtained by the model at the same location. This model is of the

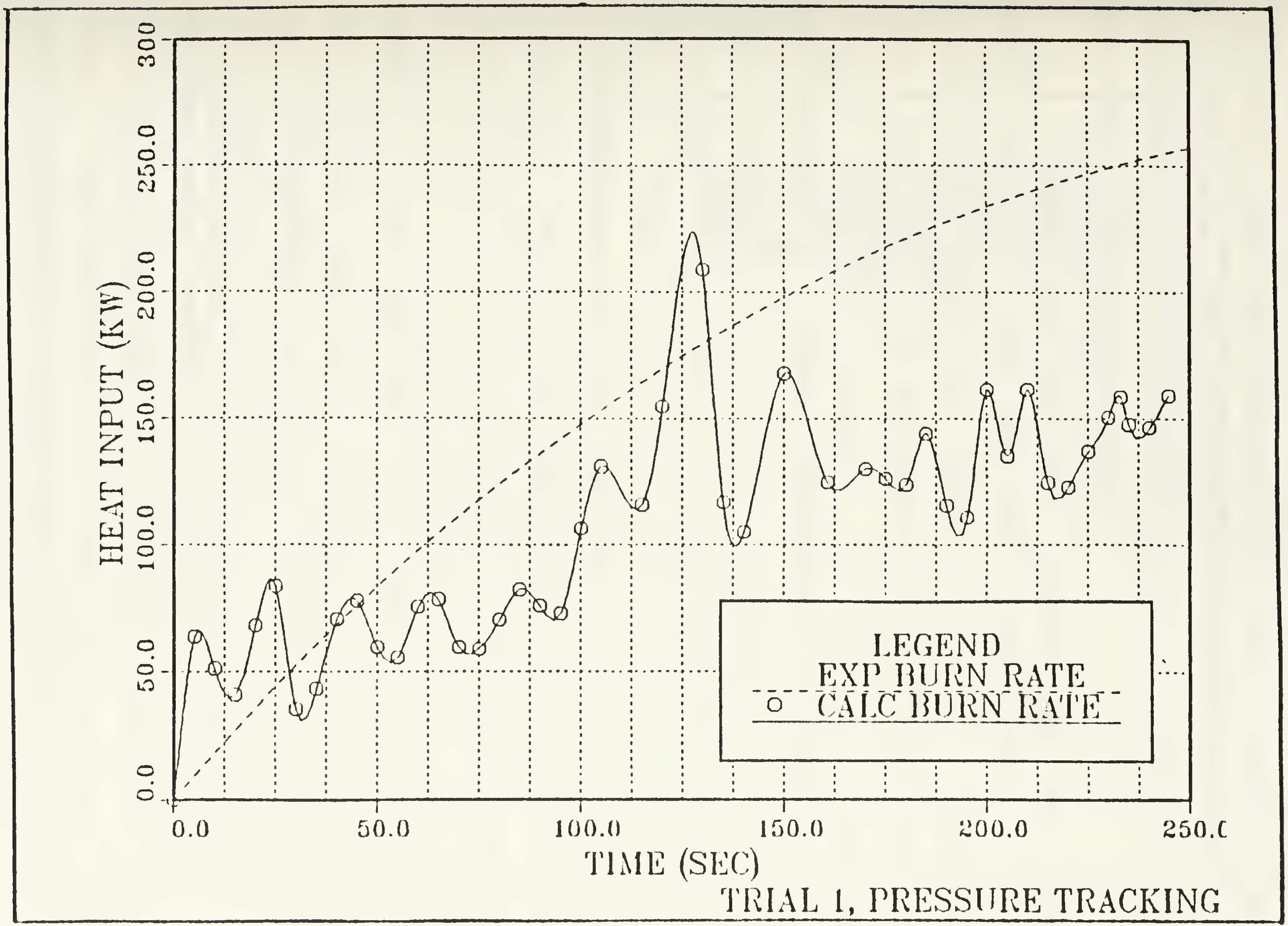


Figure 6.3 Numerical and Experimental Heat Input Curves, Trial 1

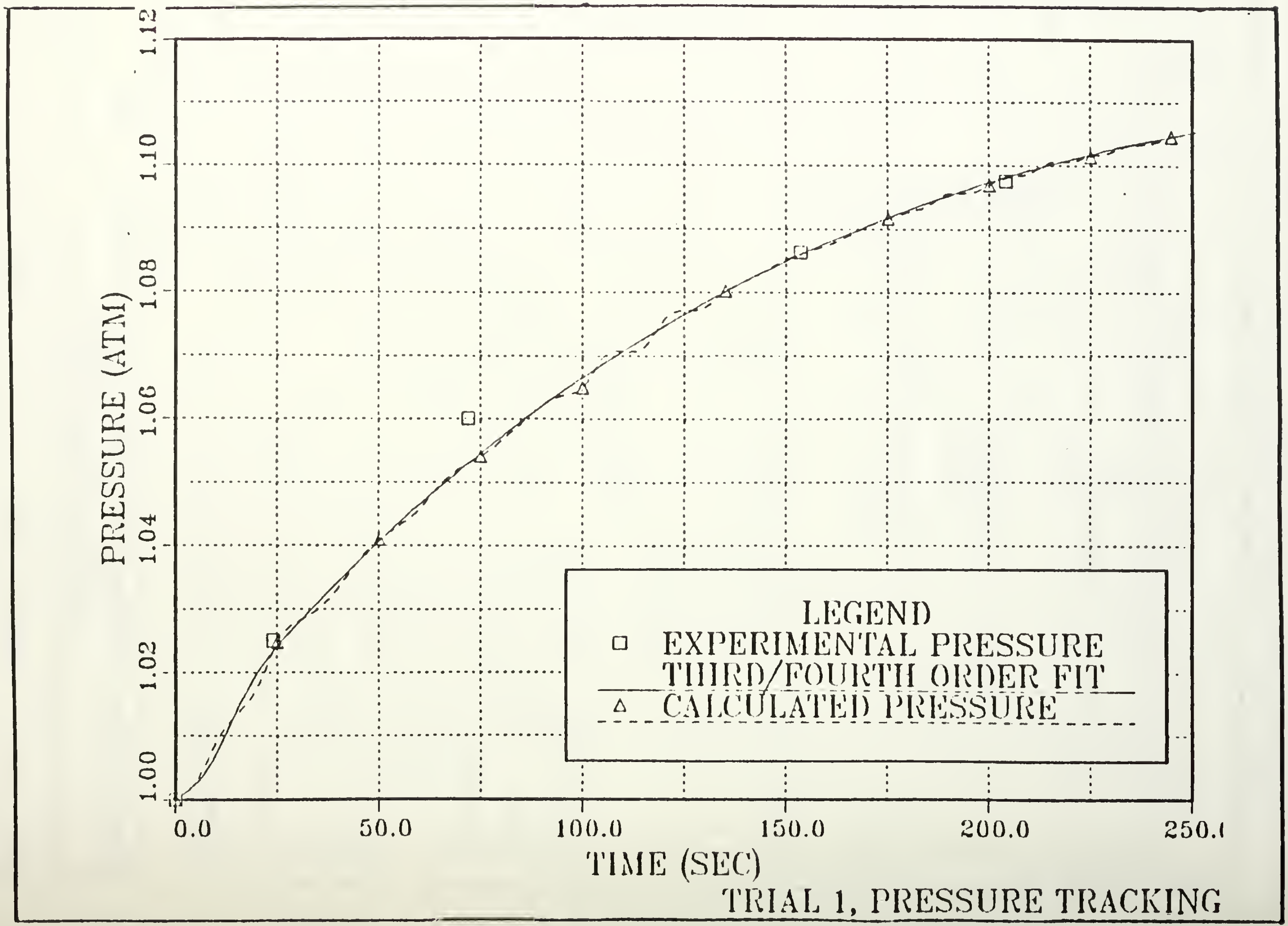


Figure 6.4 Numerical and Experimental Pressure Curves, Trial 1

same geometry as Fire-1 eliminating the requirement of finding an equivalent thermocouple location as Nies [Ref. 31] had to do. The corresponding temperature is found by interpolating the temperatures of the surrounding cells of the actual thermocouple location.

Temperatures from three thermocouples were chosen for comparison with the experimental data. These thermocouples showed the greatest change in temperature during the fire. The temperatures in the lower region of the tank changed only slightly. For this reason, they were not compared. All three thermocouples are located in the south hemispherical endcap as shown in Fig 1.1. Thermocouple 1 is located 79 inches above the midplane of the tank. Thermocouple 2 is one foot below thermocouple 1, and thermocouple 4 is two feet below thermocouple 2. The comparison between the computational and experimental results can be found in Fig 6.5-6.7. The pressure tracking routine forced the pressure to go through slight variations. Because of this, the data for the heat input rate and the corresponding temperatures amplified these variations by going through large oscillations. The computational results for all three thermocouples have the temperatures exceeding the experimental results by at least $20^{\circ}C$. The curves do tend to level off at about the same time as the experimental curves. This indicates that the same time is predicted for a quasi-steady state condition where heat in equals heat out

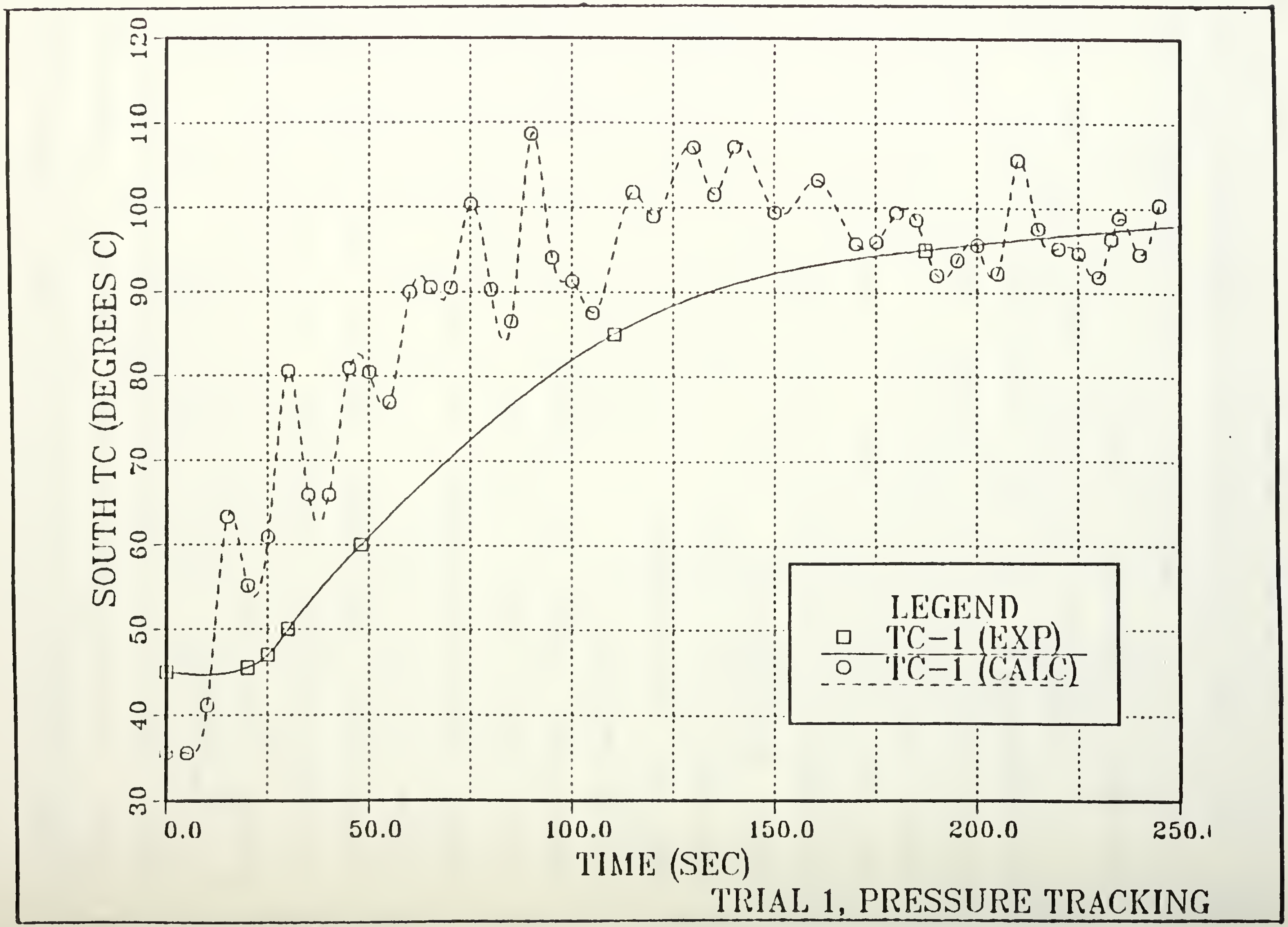


Figure 6.5 Numerical and Experimental Curves for Thermocouple #1, Trial 1

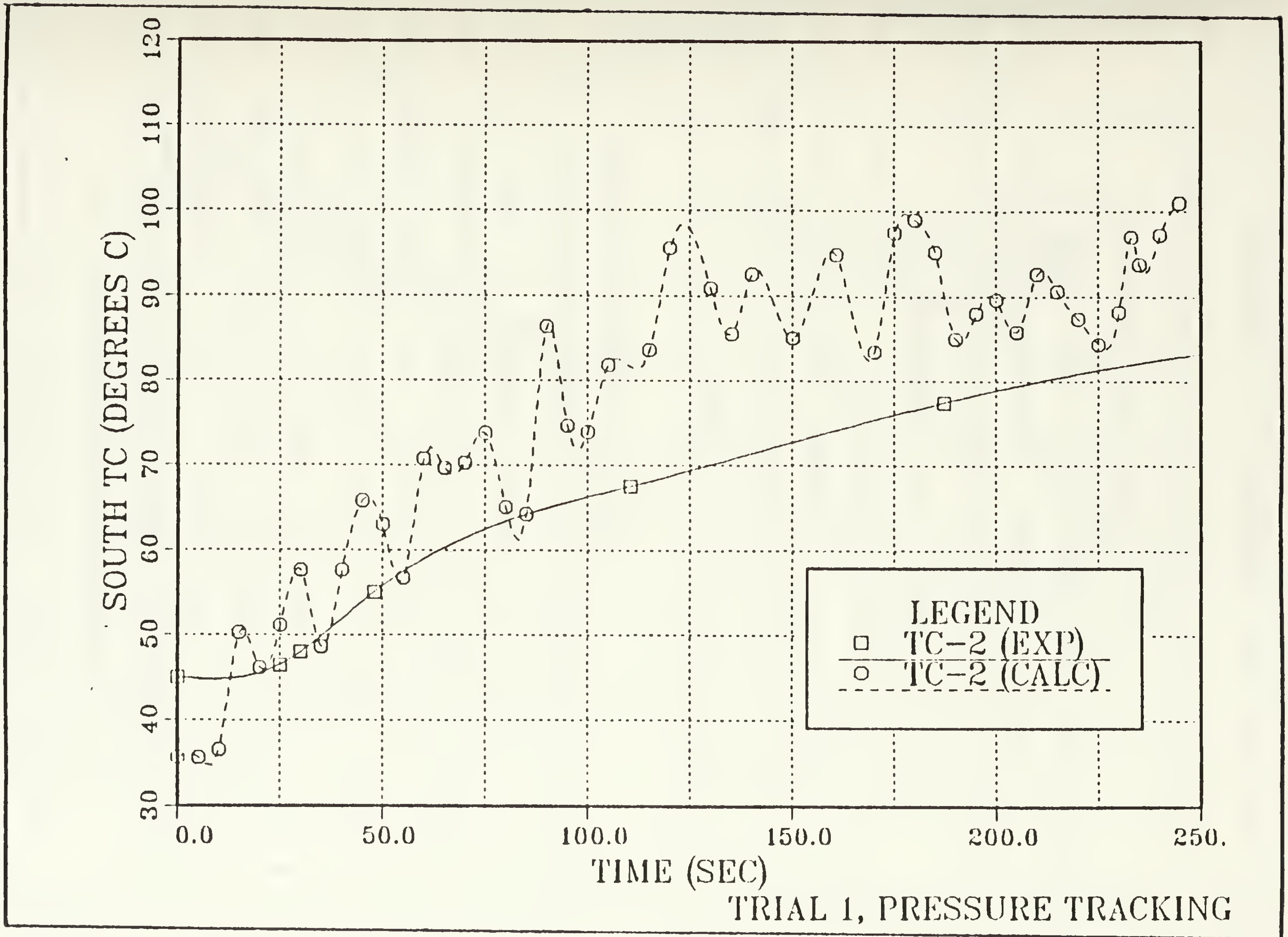


Figure 6.6 Numerical and Experimental Curves for Thermocouple #2, Trial 1

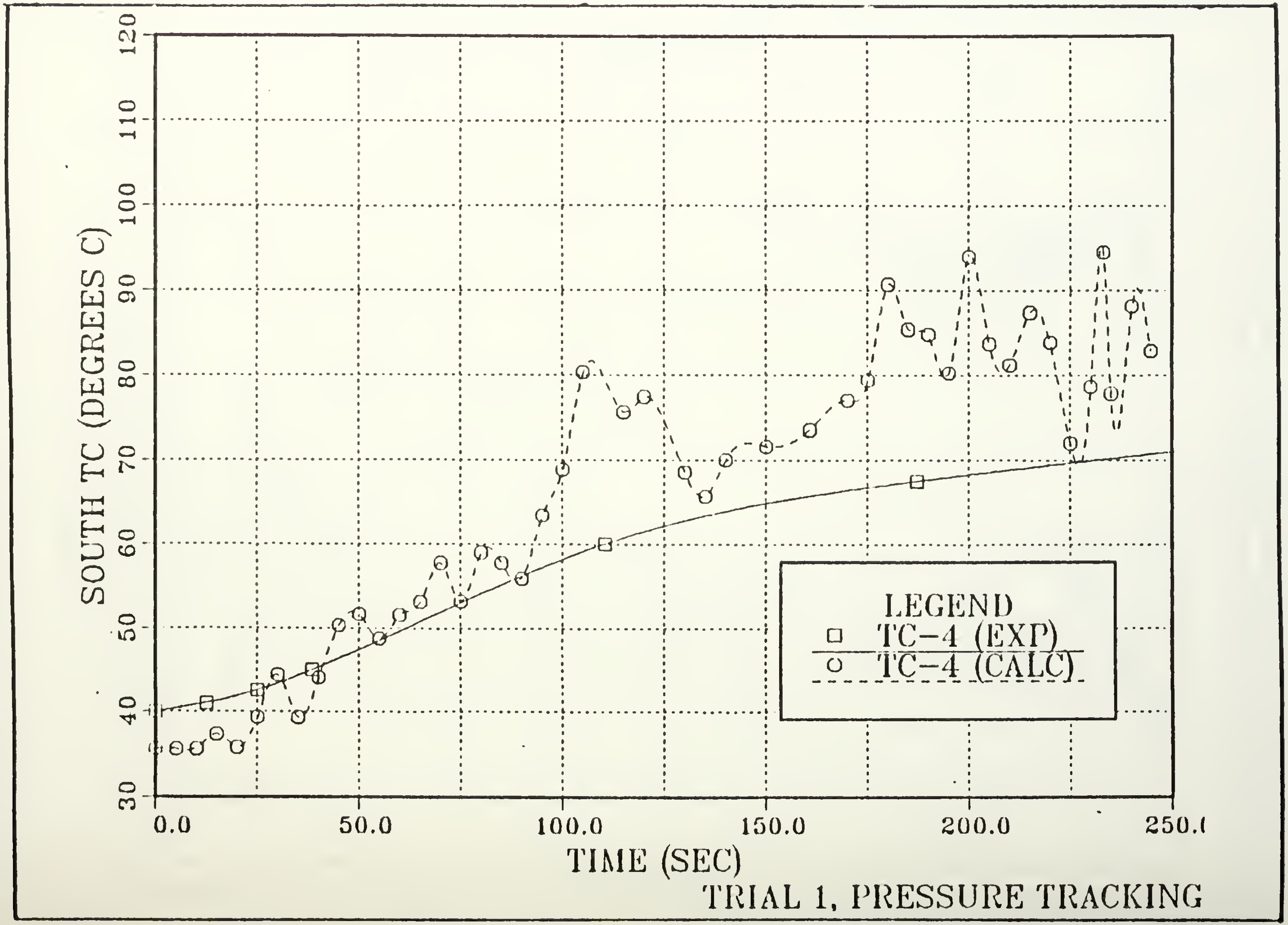


Figure 6.7 Numerical and Experimental Curves for Thermocouple #4, Trial 1

for both the model and Fire-1. The overall level of the temperature comparison is acceptable, but the oscillations are too large.

D. TRIAL 2--SIMULATED BURN RATES

NRL did provide a set of burn rate data for the methanol fire. During the experiment, there were indications of the burn rate data not being accurately recorded. The error was believed to be associated with a scaling factor. Even though the accuracy of this data was suspect, it was used to predict how pressure and temperature would respond to burn rate data input. A third order polynomial curve fit was applied through the burn rate data and entered into the program.

1. Numerical Results of Trial 2

The pressure curve, Fig. 6.8, had an initial gradual slope followed by a sharp increase in pressure. The same three thermocouple temperatures are used to compare the experimental data to the computer model. The temperature curves are found in Figs. 6.9-6.11. The computational temperature initially followed the experimental curve, but then overshot the experimental readings by a factor of two for thermocouples 1 and 2. Thermocouple 4 had temperature readings below the experimental temperature, but the numerical result was beginning to show an increase in slope at the end of the computer run. Both temperature and pressure did not show any sign of leveling off, indicating

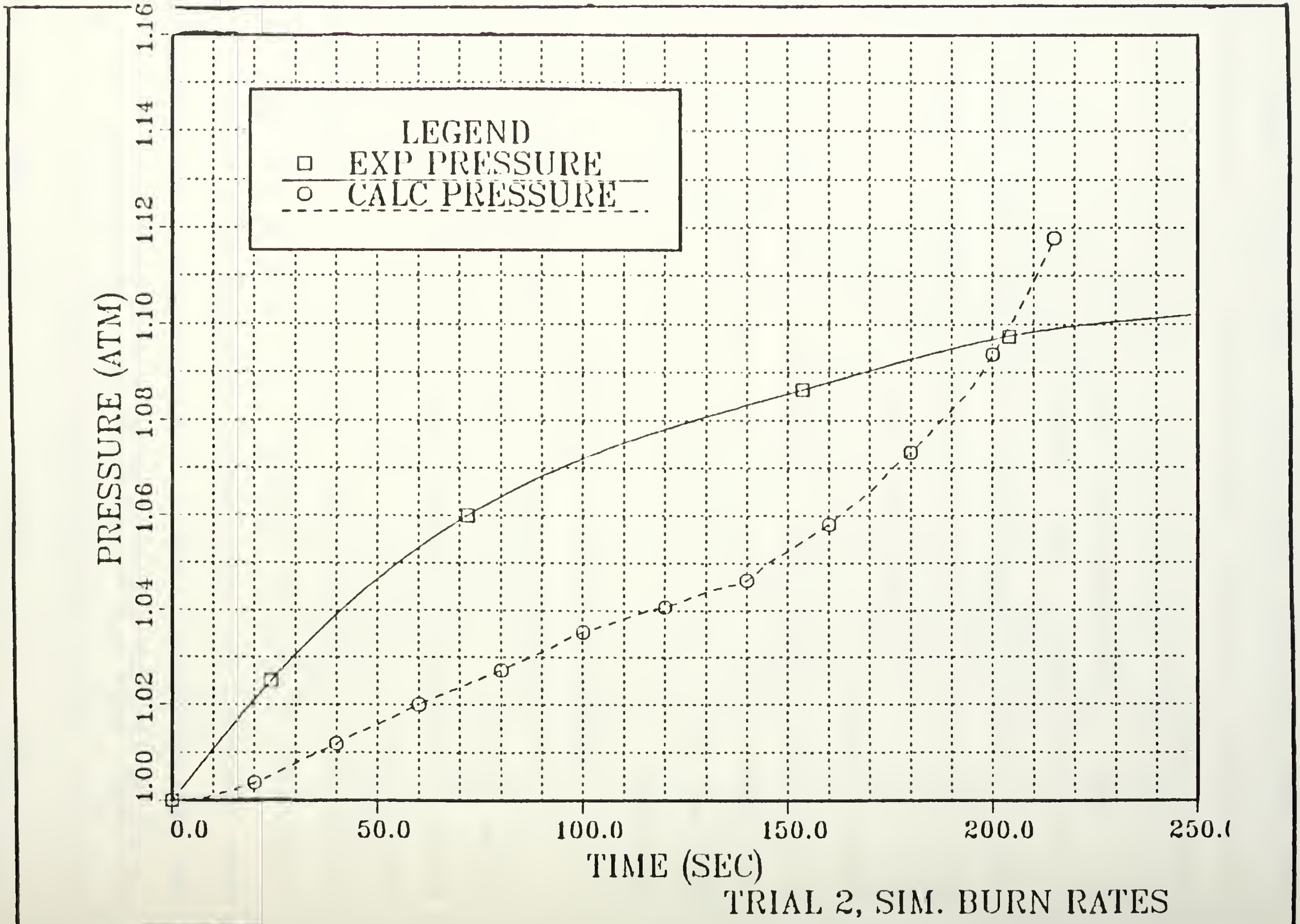


Figure 6.8 Numerical and Experimental Pressure Curves, Trial 2

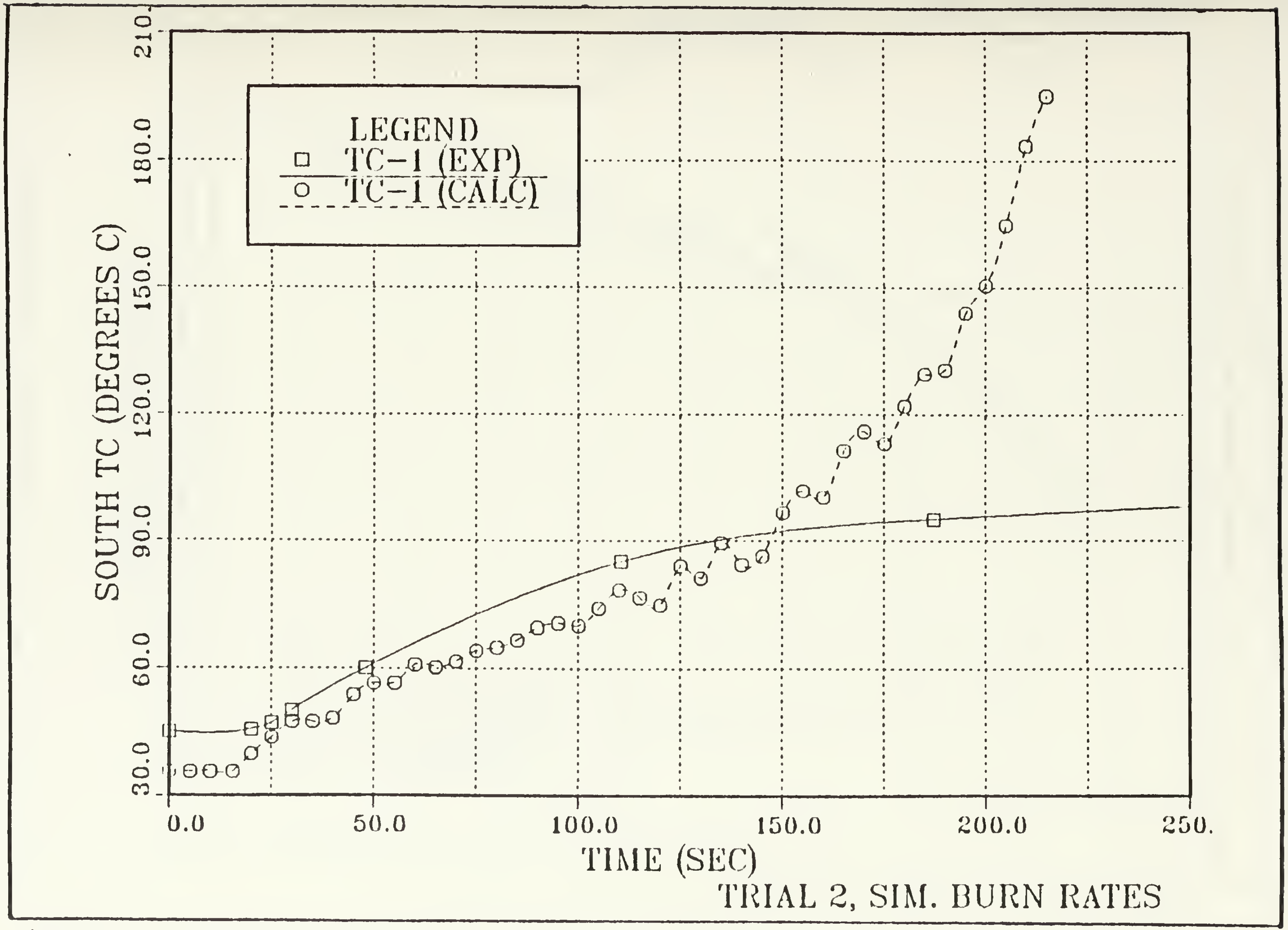


Figure 6.9 Numerical and Experimental Curves for Thermocouple #1, Trial 2

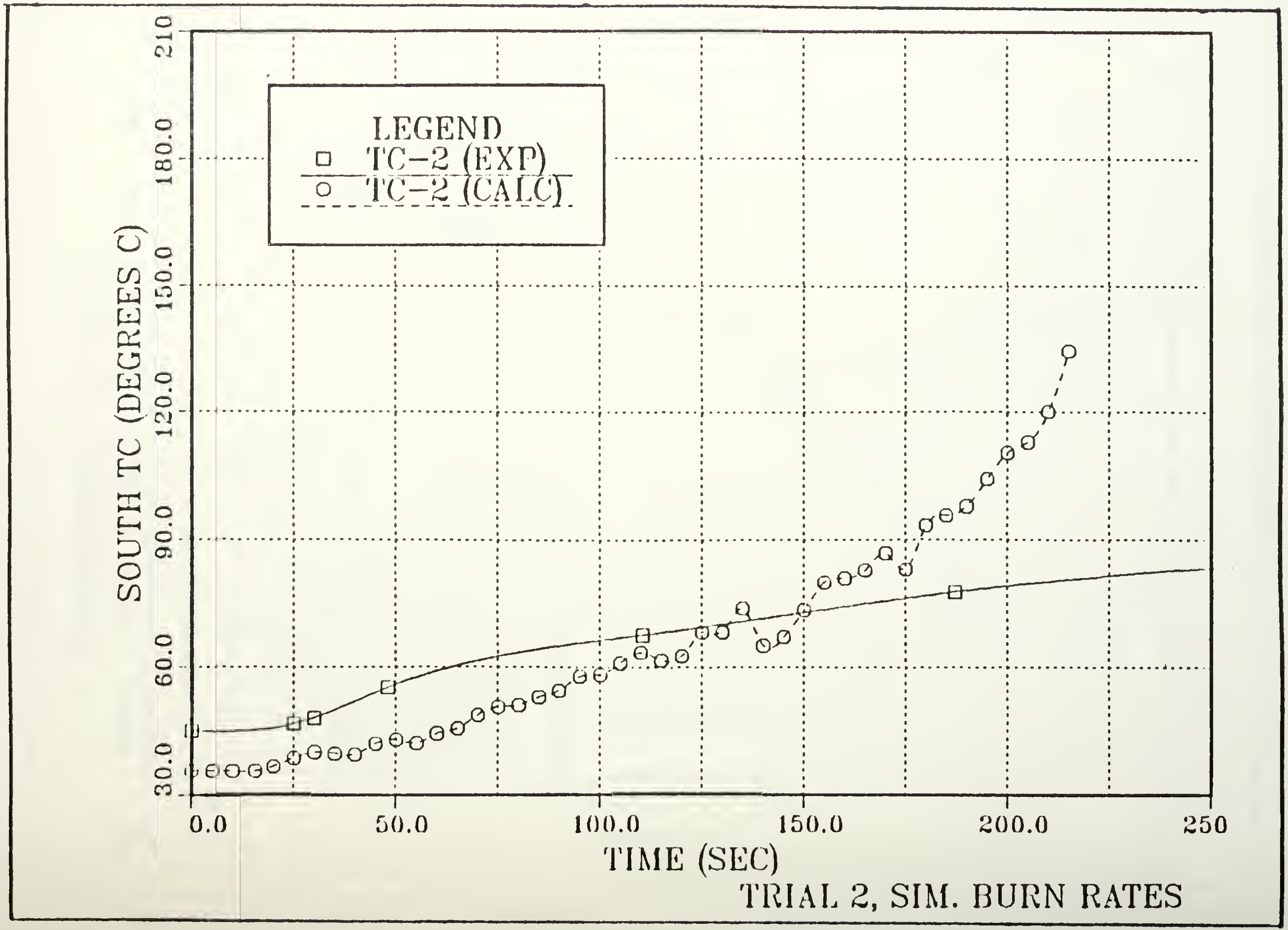


Figure 6.10 Numerical and Experimental Curves for Thermocouple #2, Trial 2

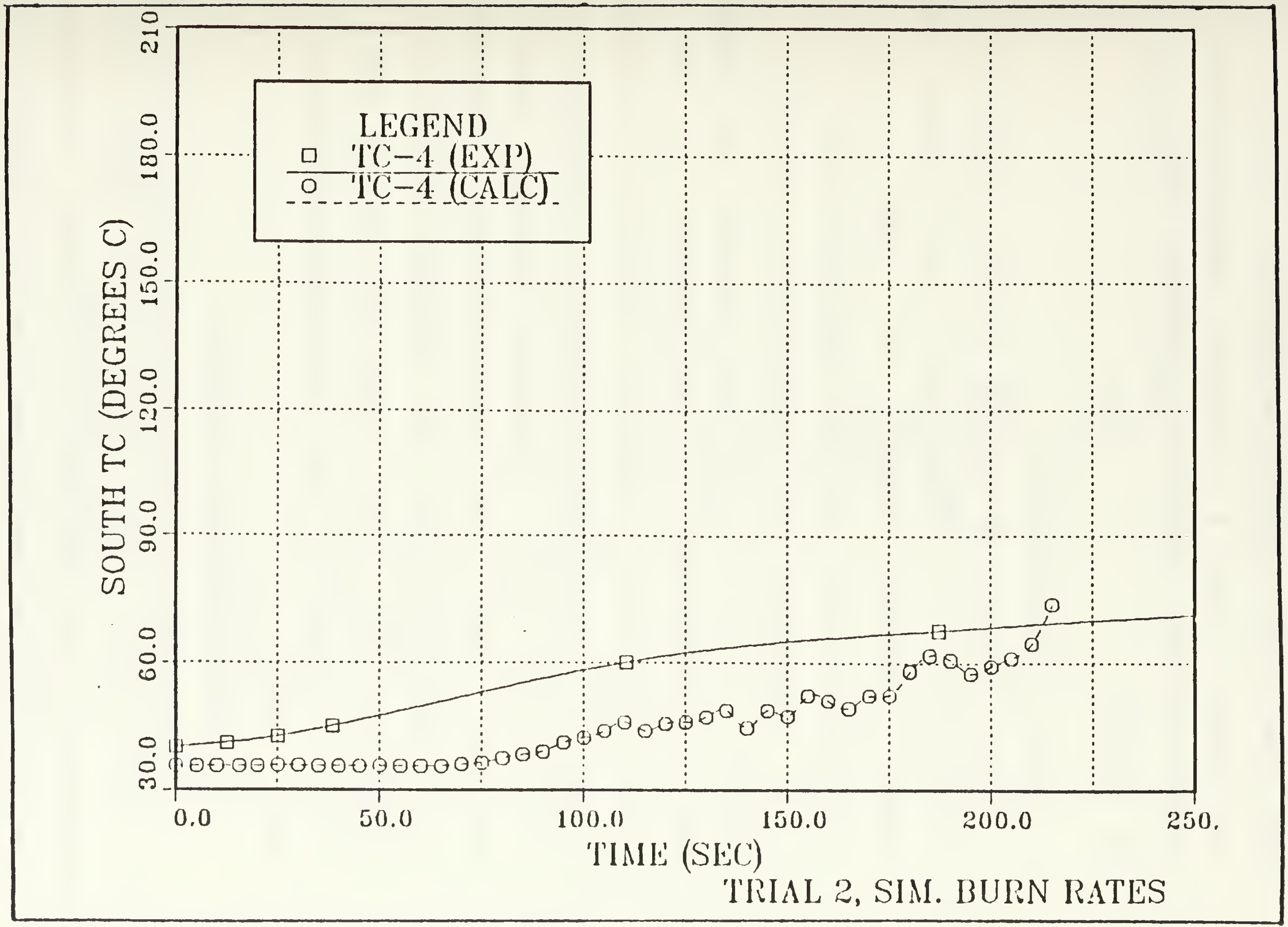


Figure 6.11 Numerical and Experimental Curves for Thermocouple #4, Trial 2

the heat input curve was too high. By applying a steady heat input curve, the temperature and pressure did not oscillate as in Trial 1. Therefore, oscillations were not a problem, but the levels for temperature and pressure were not correct.

D. TRIAL 3--COMBINATION

After reviewing the results of Trial 1 and Trial 2, it was decided to use a combination of the two cases. Trial 1 involved a pressure tracking routine known to cause large oscillations in heat input and temperature curves. In Trial 2, the experimental burn rate used was too high. This resulted in exceeding experimental pressure and temperature data. The combination case will use the burn rate data generated from Trial 1. However, this data will be modified by applying a third order polynomial fit through the data points (see Fig. 6.12). This gives a burn rate curve of the same form as Trial 2, but modified in terms of magnitude to allow comparison between the numerical results and the experimental data. Using a burn rate curve as input, both temperature and pressure are available for validation of the computer code.

1. Numerical Results of Trial 3

The pressure of the tank over 130 sec of fire time is found in Fig. 6.13. Unfortunately, due to the large amount of CPU time required to run one case, the shorter run time expressed here was all that could be accomplished for

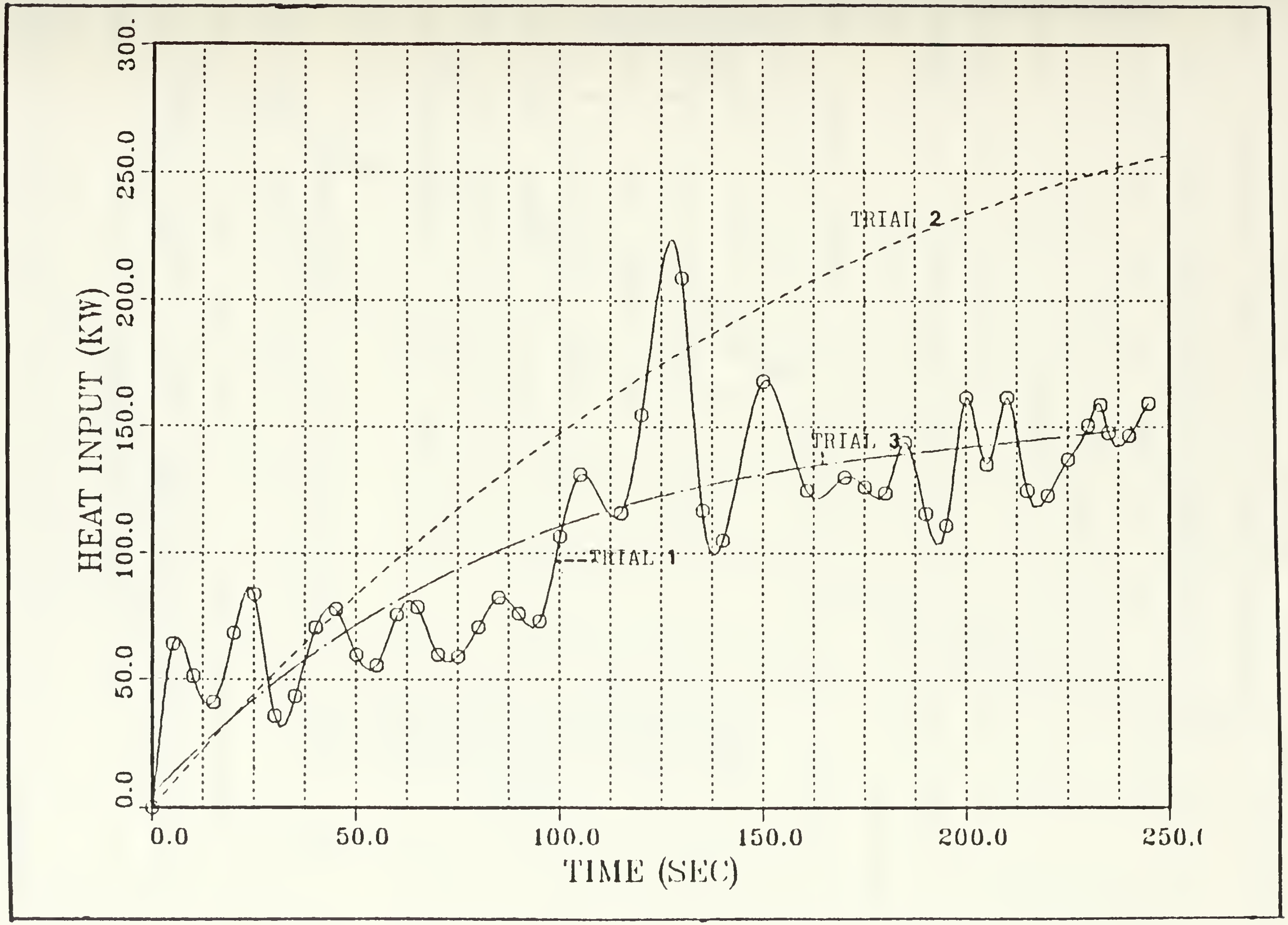


Figure 6.12 Numerical and Experimental Heat Input Curves, Trial 1, 2, and 3

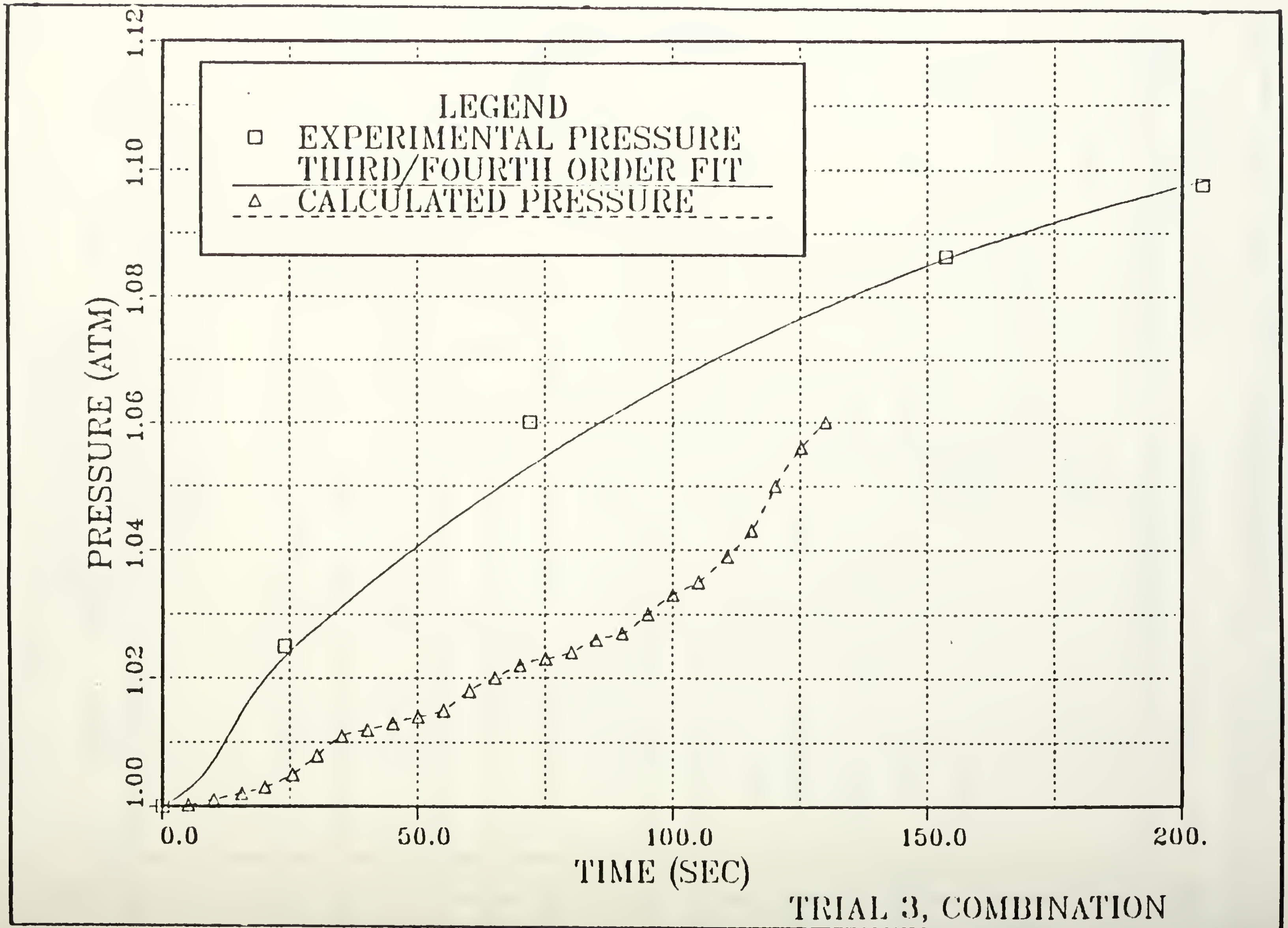


Figure 6.13 Numerical and Experimental Pressure Curves, Trial 3

this thesis. Early indications show the pressure obtained by the computer code fall below the experimental curve.

The temperature comparison for the three thermocouples can be found in Figs. 6.14-6.16. The temperatures also show slight oscillations, but not to the extent as Trial 1. Thermocouple 1 tracked closely to the experimental temperatures. It is interesting to note the experimental temperature did start at a higher temperature. The initial assumption used in the computer code was for all the temperatures in the tank to equal the ambient temperature, which was 35.6^{o}C on the day of the test. However, by extracting the data from the curves provided by NRL, Fig. 6.17, the initial temperature of the thermocouples varied. This could be due to a slight internal heating of Fire-1 by external means, i.e., the sun. Fire-1 is not enclosed inside a building. If the initial temperature of the computed thermocouple readings were increased to match the experimental data, the curves would show a closer correlation.

Thermocouple 4 has experimental points that are extremely hard to read in the initial few minutes of the test run. This could result in a misrepresentation of the experimental curve. The experimental temperature curves used for comparison between the computer model are generated from applying a smooth curve through a few experimental points from Fig. 6.17. Actually the experimental

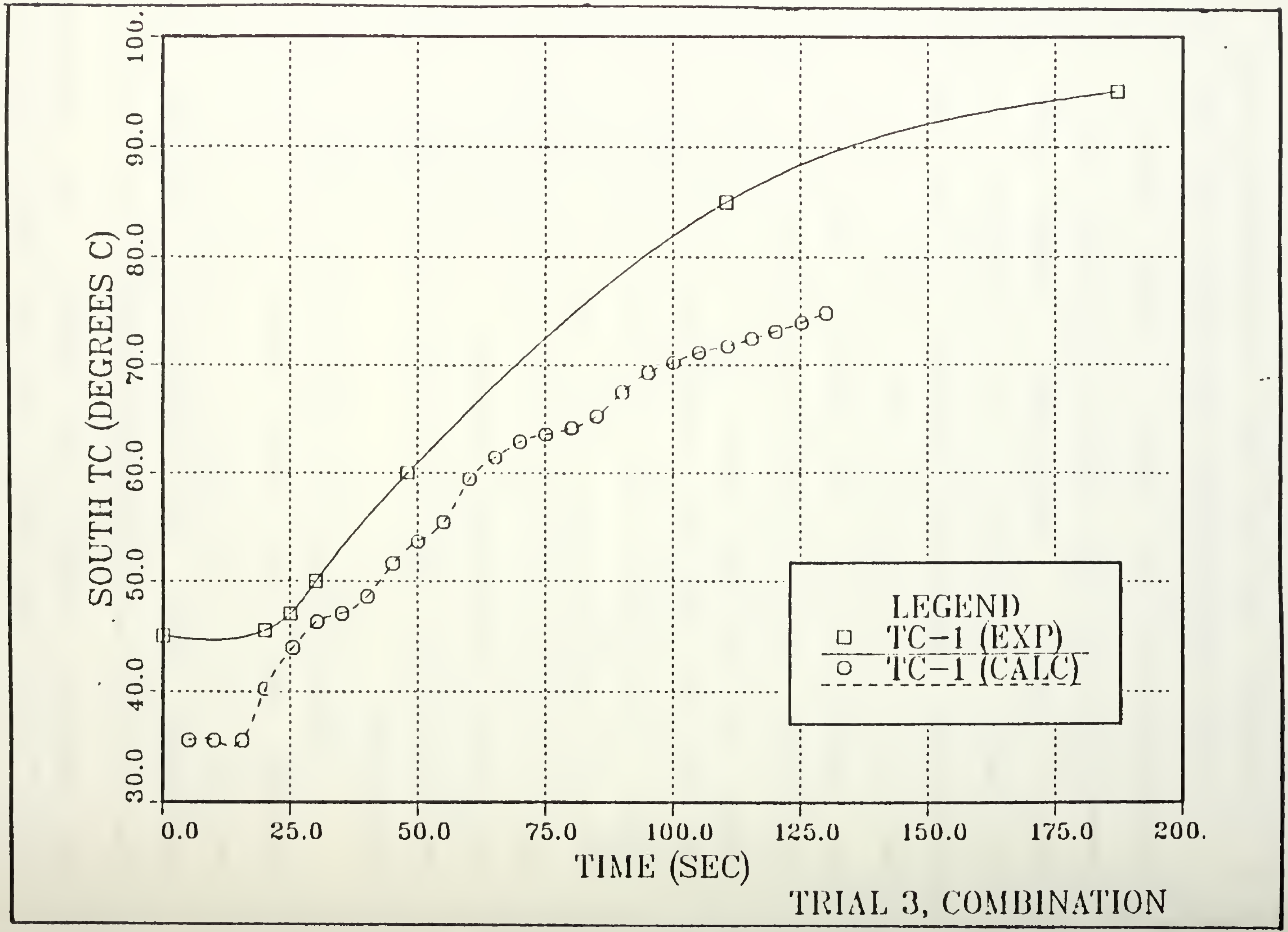


Figure 6.14 Numerical and Experimental Curves for Thermocouple #1, Trial 3

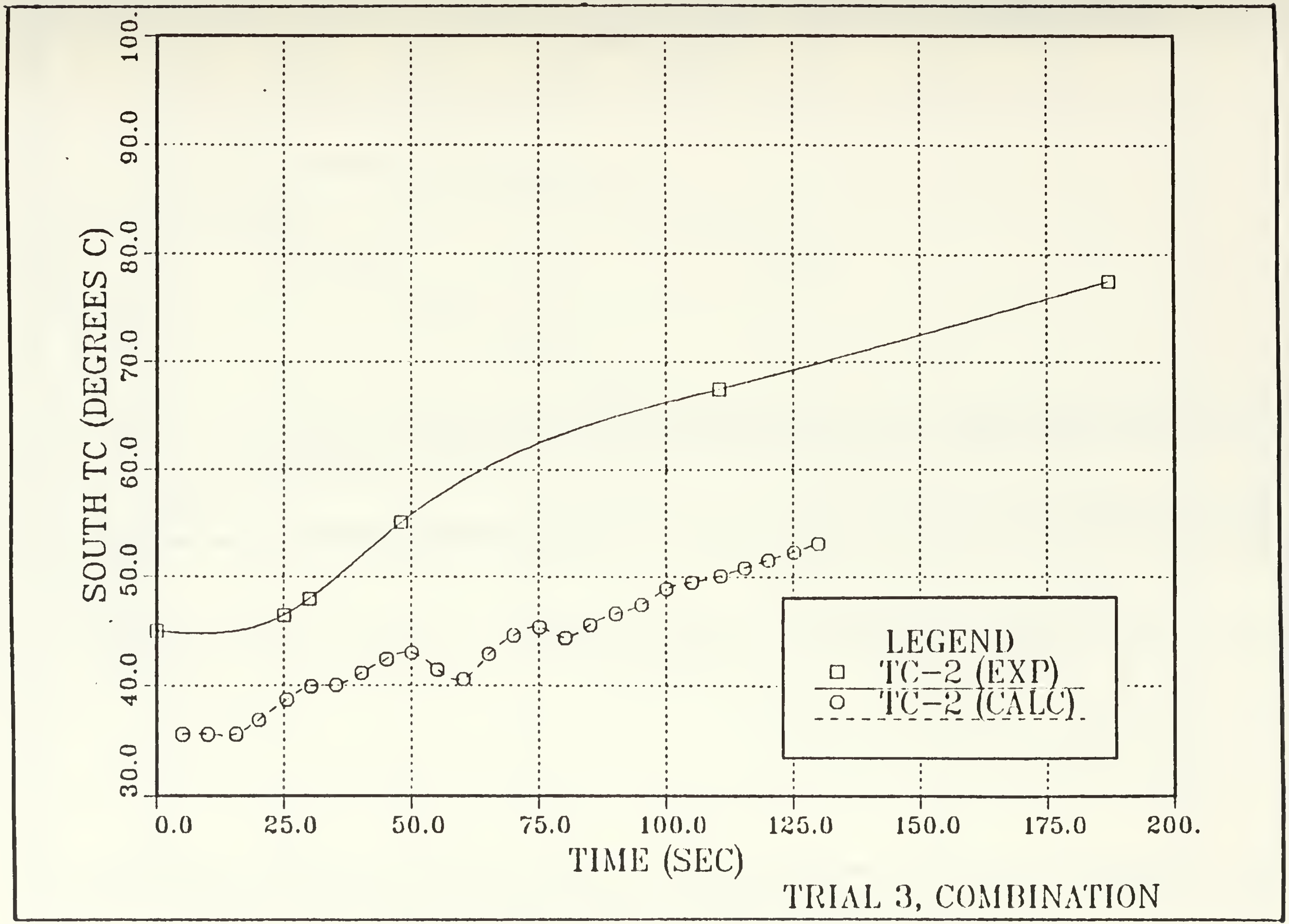


Figure 6.15 Numerical and Experimental Curves for Thermocouple #2, Trial 3

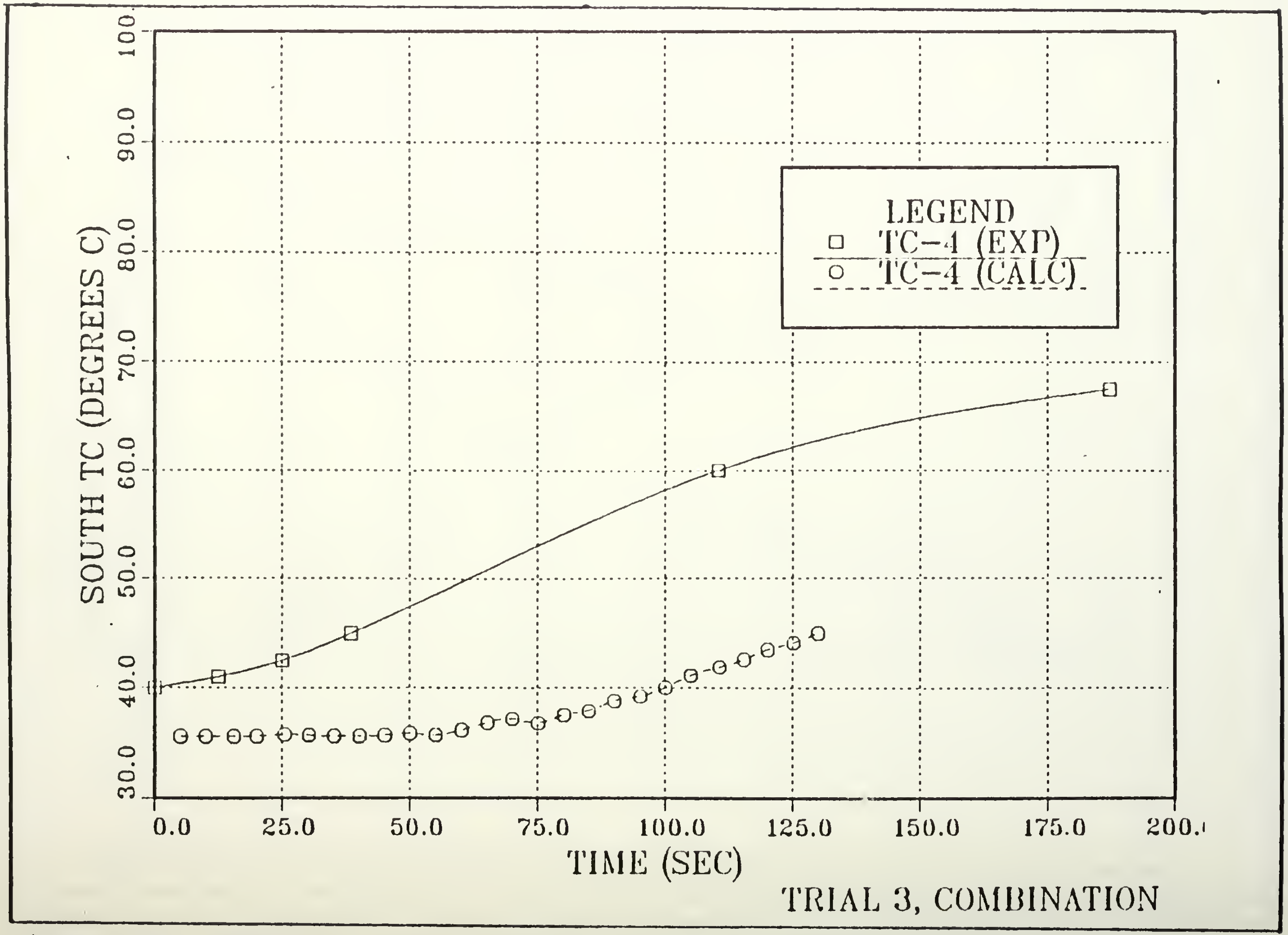


Figure 6.16 Numerical and Experimental Curves for Thermocouple #4, Trial 3

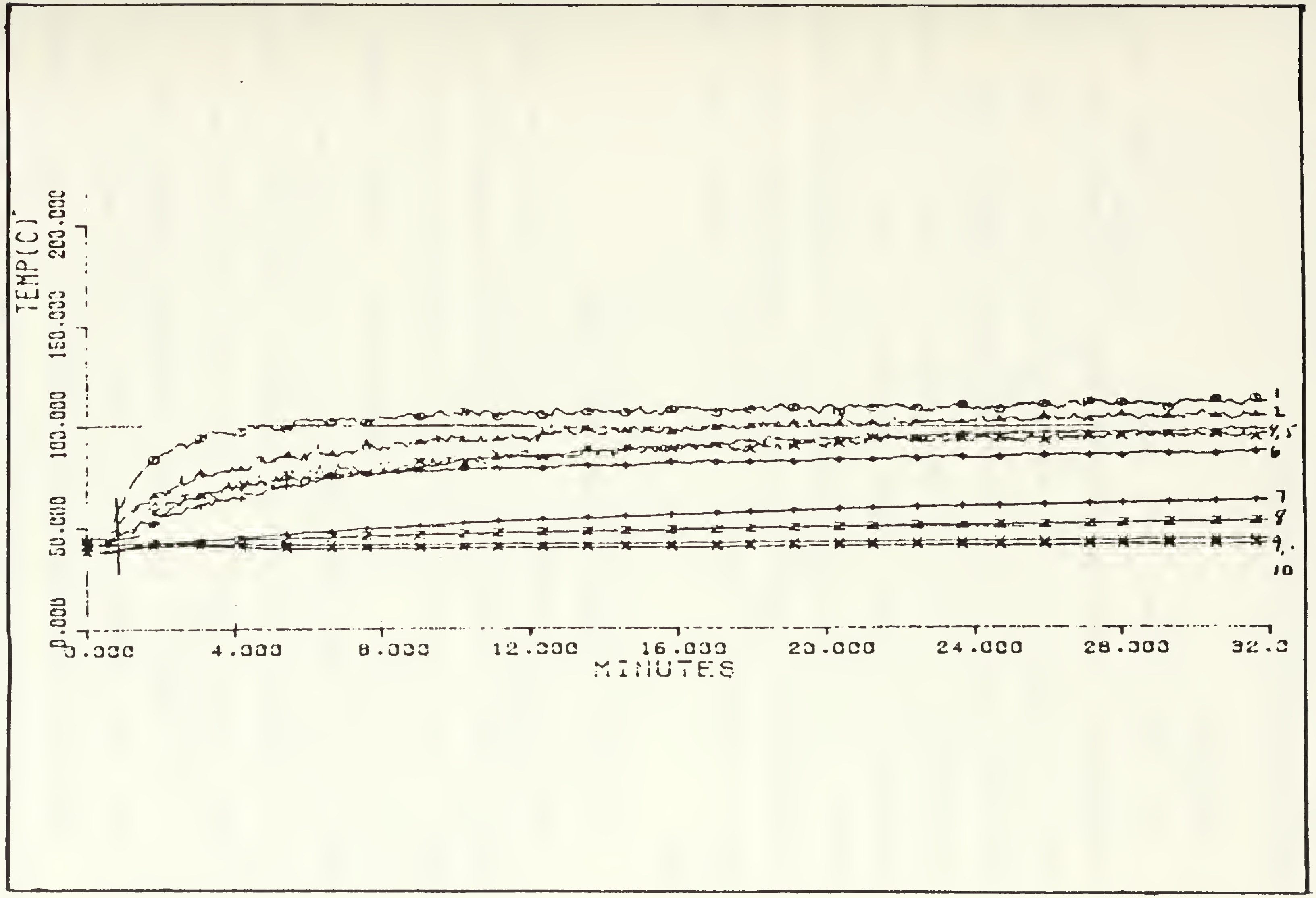


Figure 6.17 Experimental Temperatures Provided by NRL for Thermocouples 1-10 Located at South Rack

temperatures do have slight oscillations and uncertainty values associated with them.

F. RESULTS

Of the three test cases, Trial 3 has the best correlation to the experimental data. However, this was the result of combining the two previous trials. This further amplifies the need for accurate heat release data. It is not practical to do an entire computer simulation using the principles of Trial 1 just to obtain a heat release curve. And then use this curve as input to run the actual case. It is necessary to have the required experimental heat release data as input in order to accurately assess the computer code.

G. VELOCITY PROFILE AND ISOTHERM PLOTS

The velocity profile and isotherm plots are given for Trial 3 since this trial is the best representation of the actual fire in Fire-1. Fig. 6.18 shows the location of the two-dimensional cross sectional areas chosen for examination. The three representative locations are located at the midplane through the center of the tank and through a circular cross section at the fire and thermocouple rack. Fig 6.19 through 6.30 show the three planar views of the velocity and isotherm plots at 30, 60, 90, and 130 sec.

These are two dimensional plots of a three dimensional model. This is the only means available at this time to

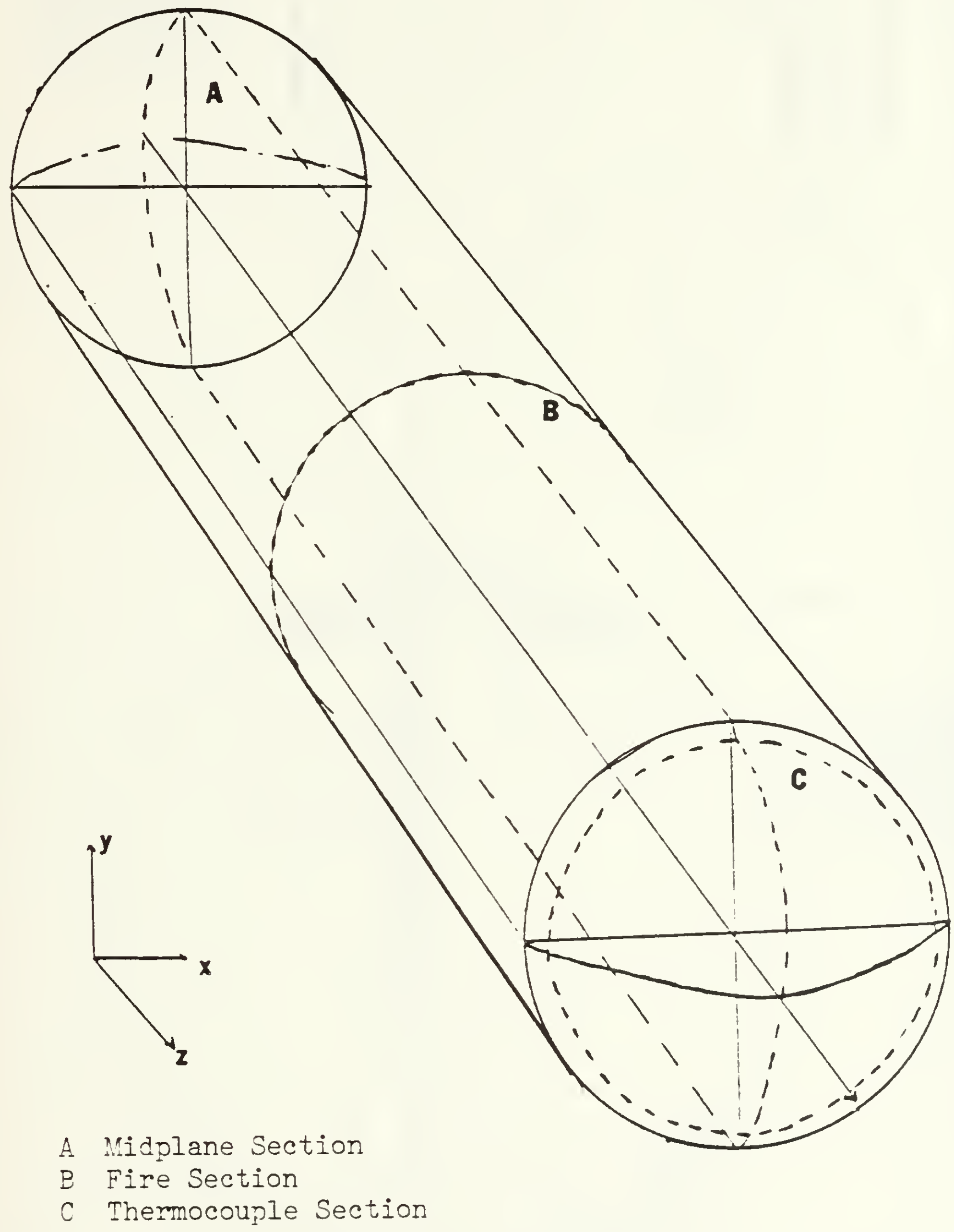


Figure 6.18 Location of Cross Sections used for Isotherm and Velocity Plots

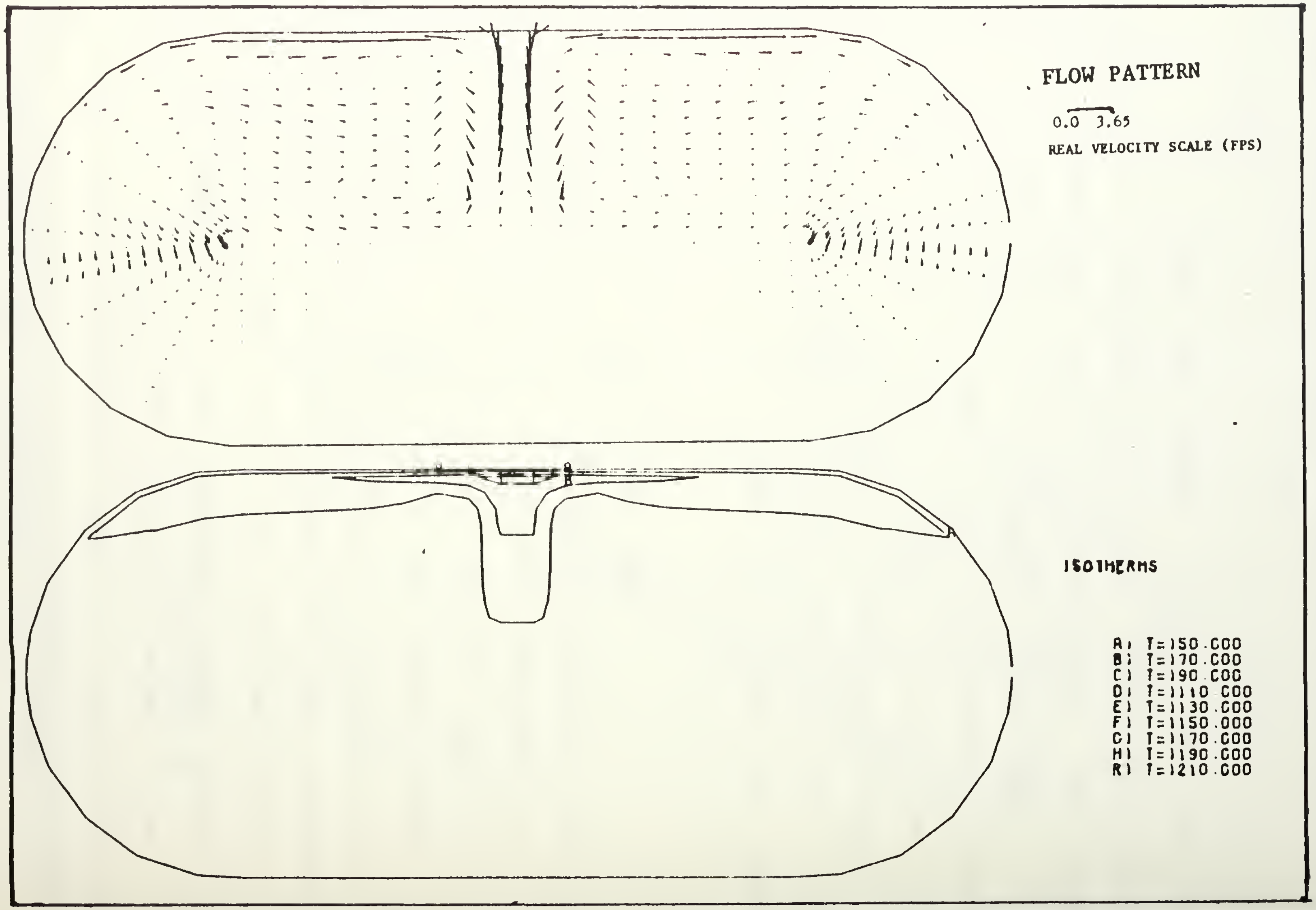


Figure 6.19 Velocity and Isotherm Plots after 30 Sec at Midplane

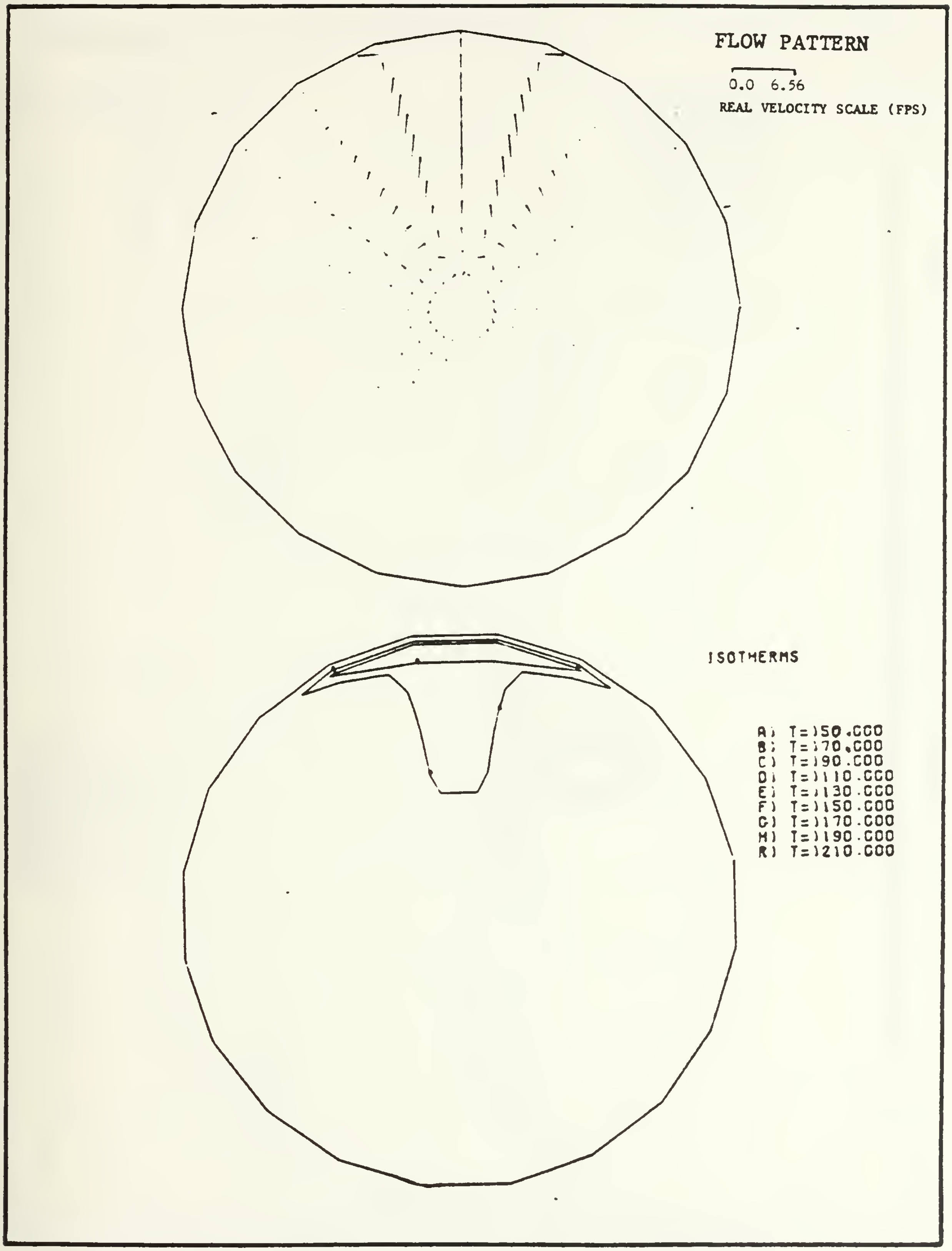


Figure 6.20 Velocity and Isotherm Plots after 30 Sec at Fire Center

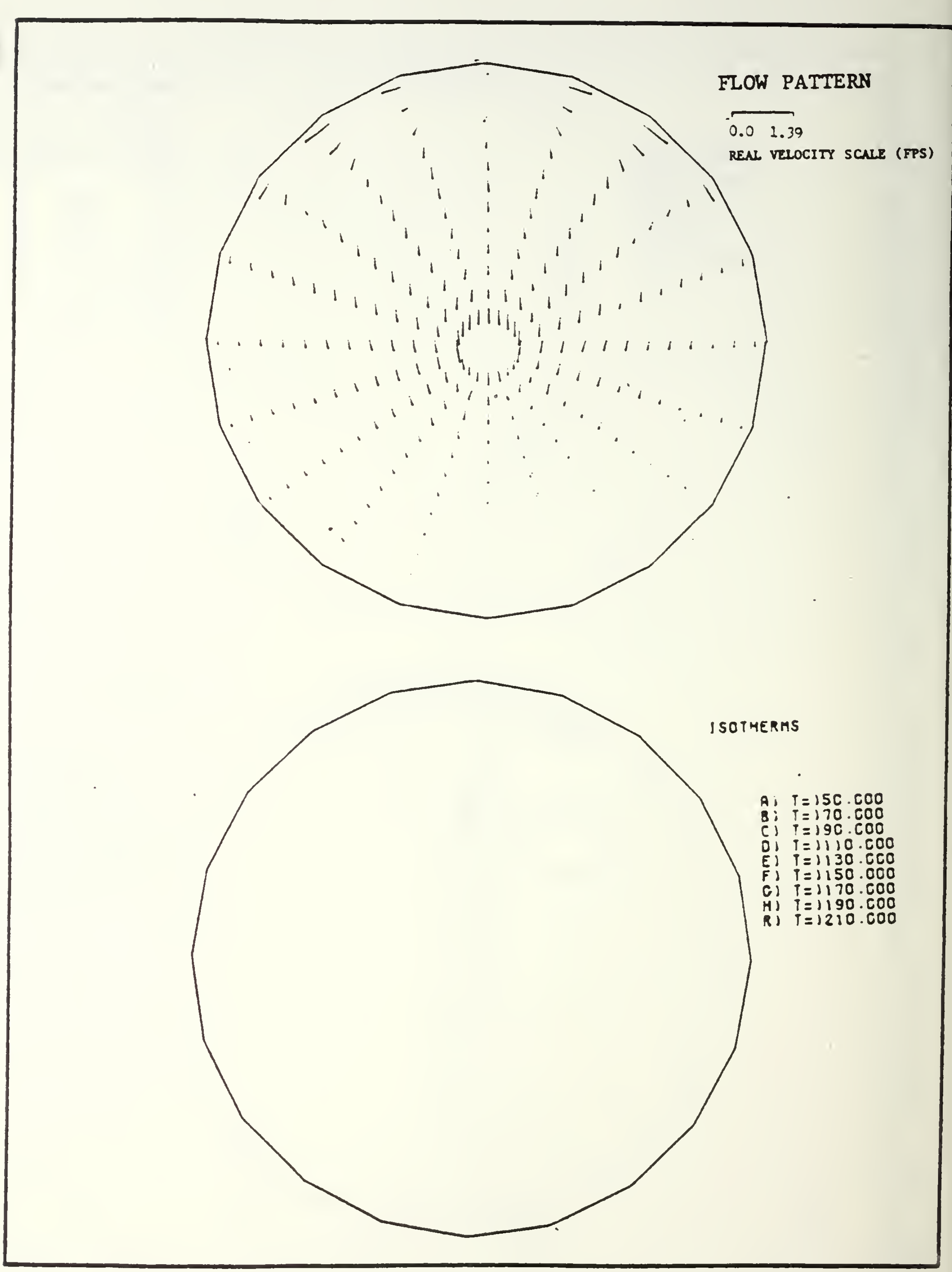


Figure 6.21 Velocity and Isotherm Plots after 30 Sec at the Thermocouple Rack

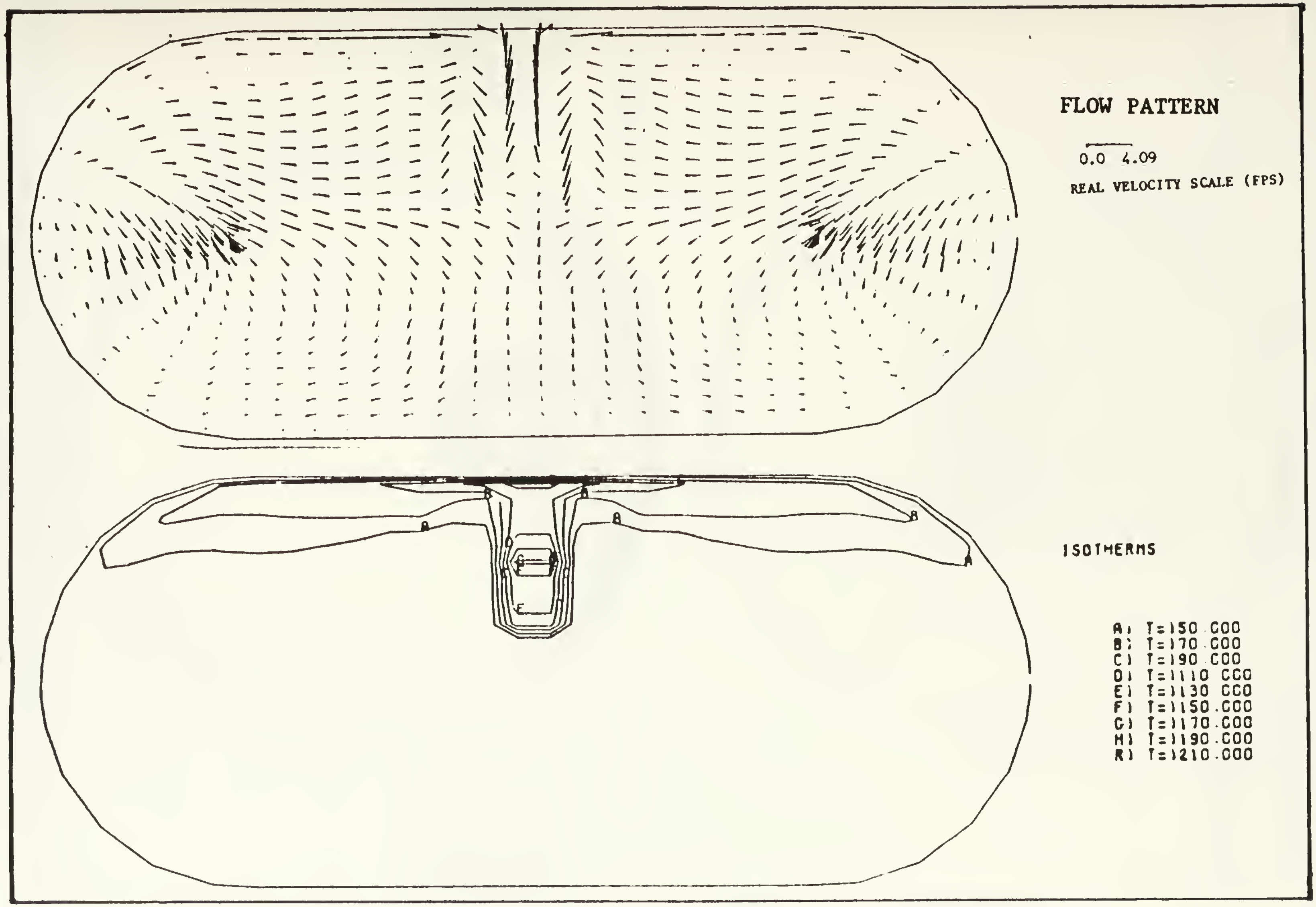


Figure 6.22 Velocity and Isotherm Plots after 60 Sec at Midplane

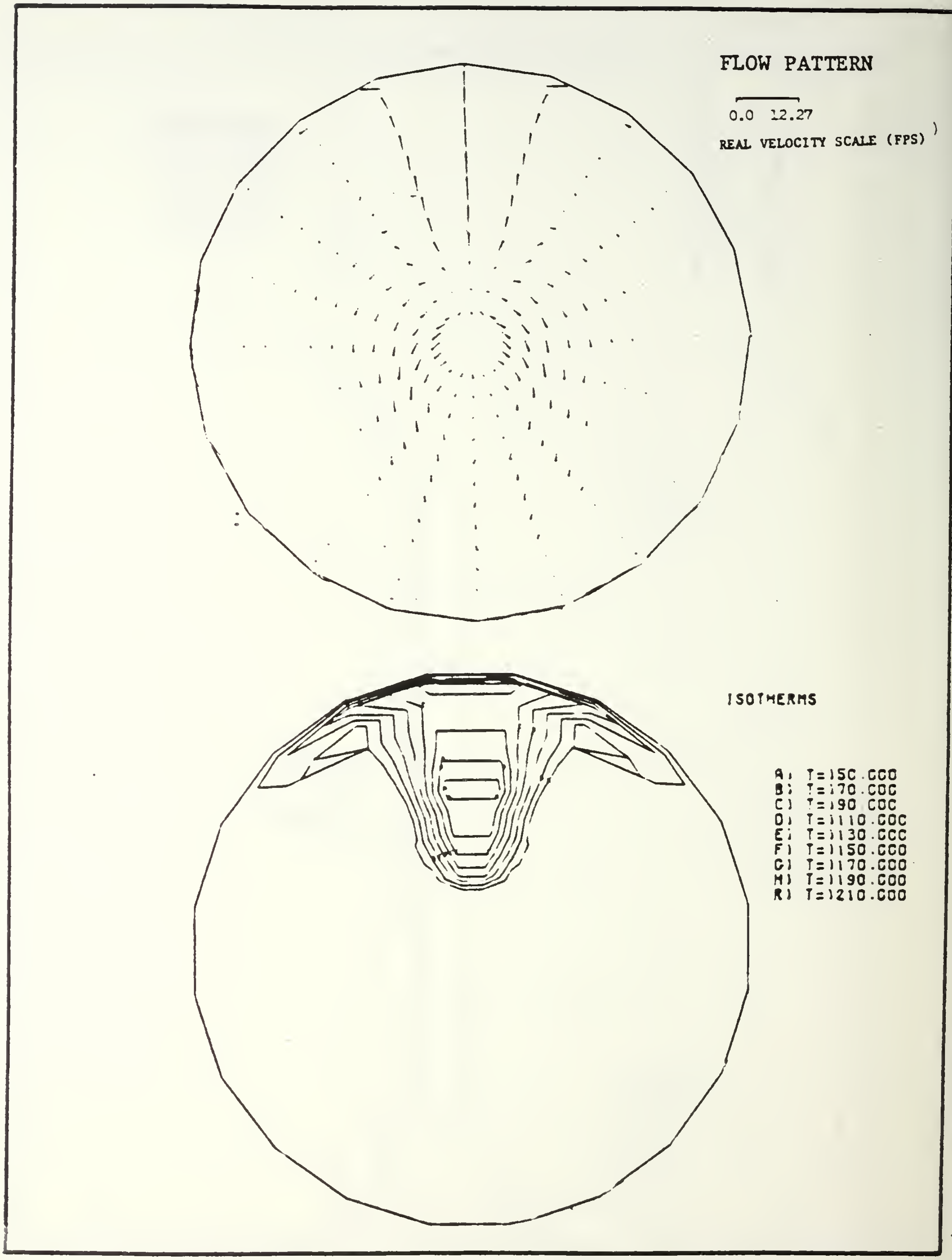


Figure 6.23 Velocity and Isotherm Plots after 60 Sec at Fire Center

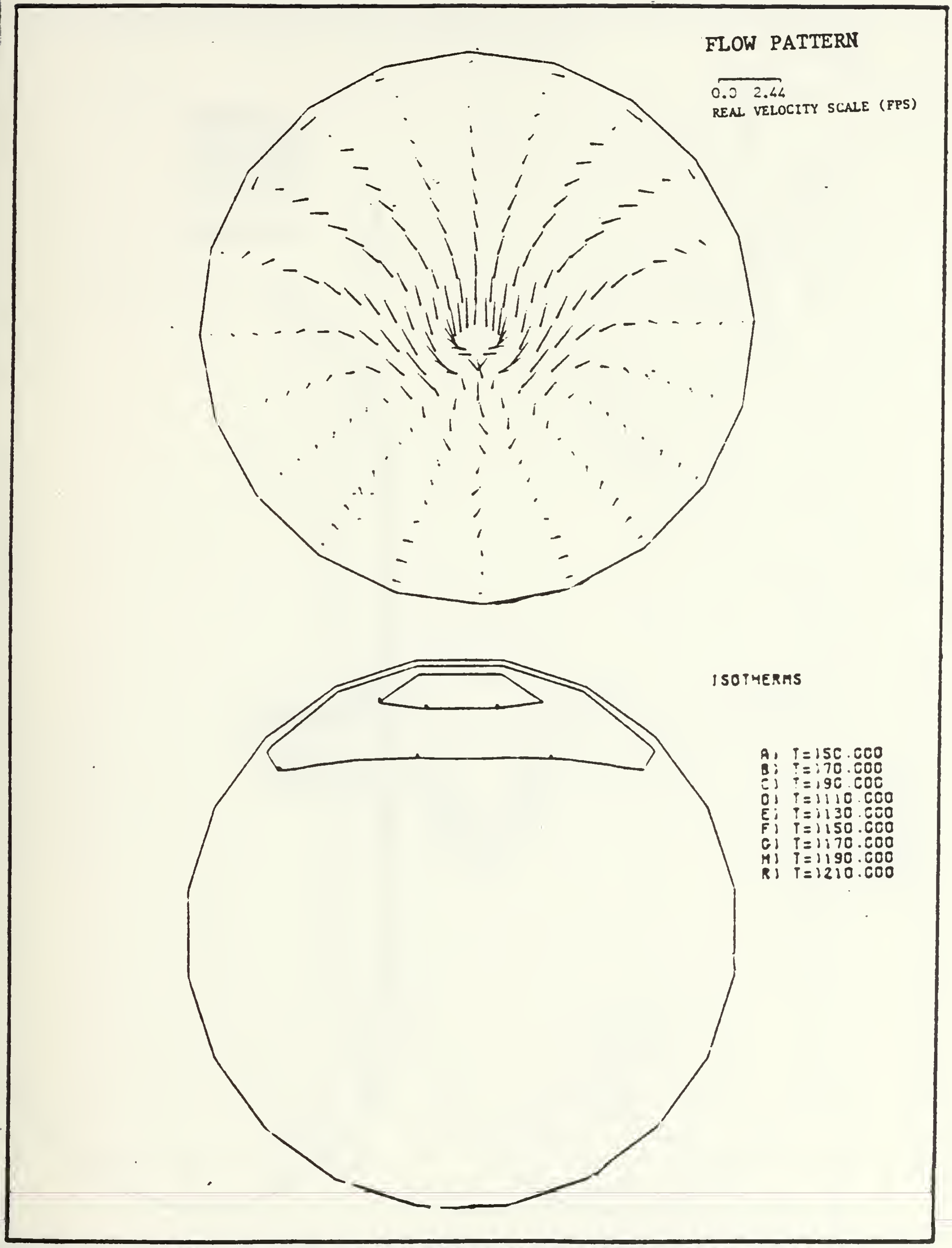


FIgure 6.24 Velocity and Isotherm Plots after 60 Sec at the Thermocouple Rack

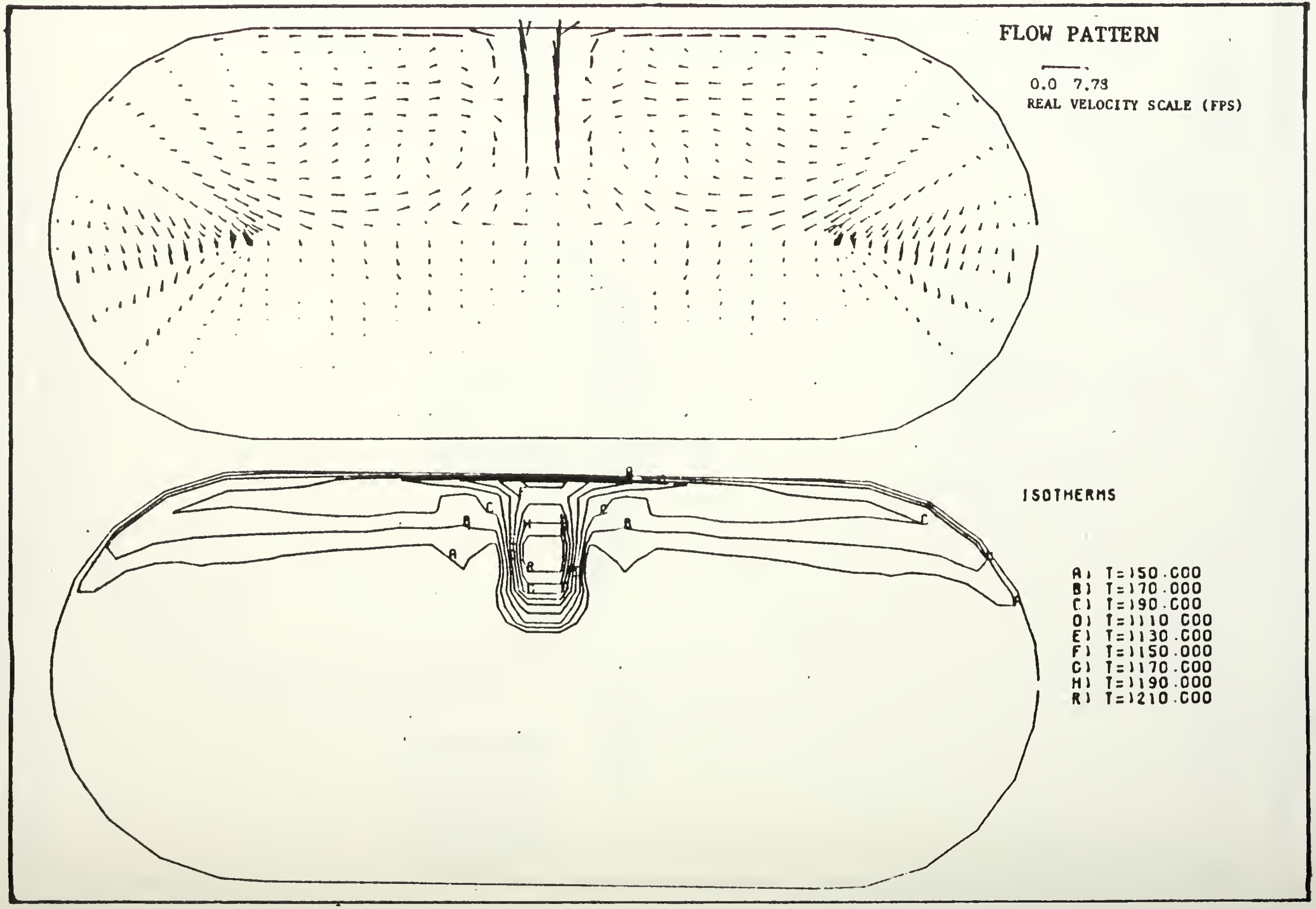


Figure 6.25 Velocity and Isotherm Plots after 90 Sec at Midplane

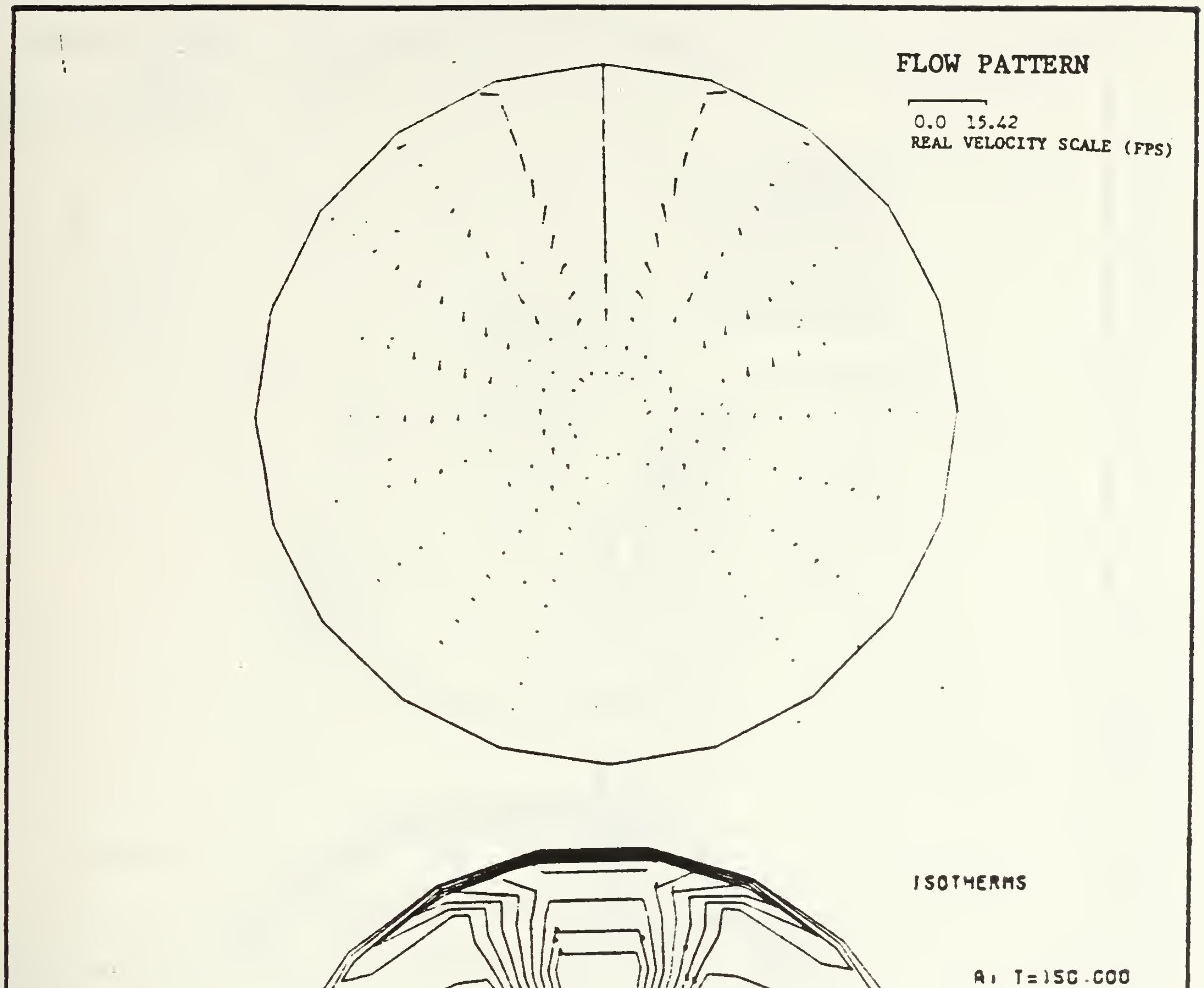


Figure 6.26 Velocity and Isotherm Plots after 90 Sec at Fire Center

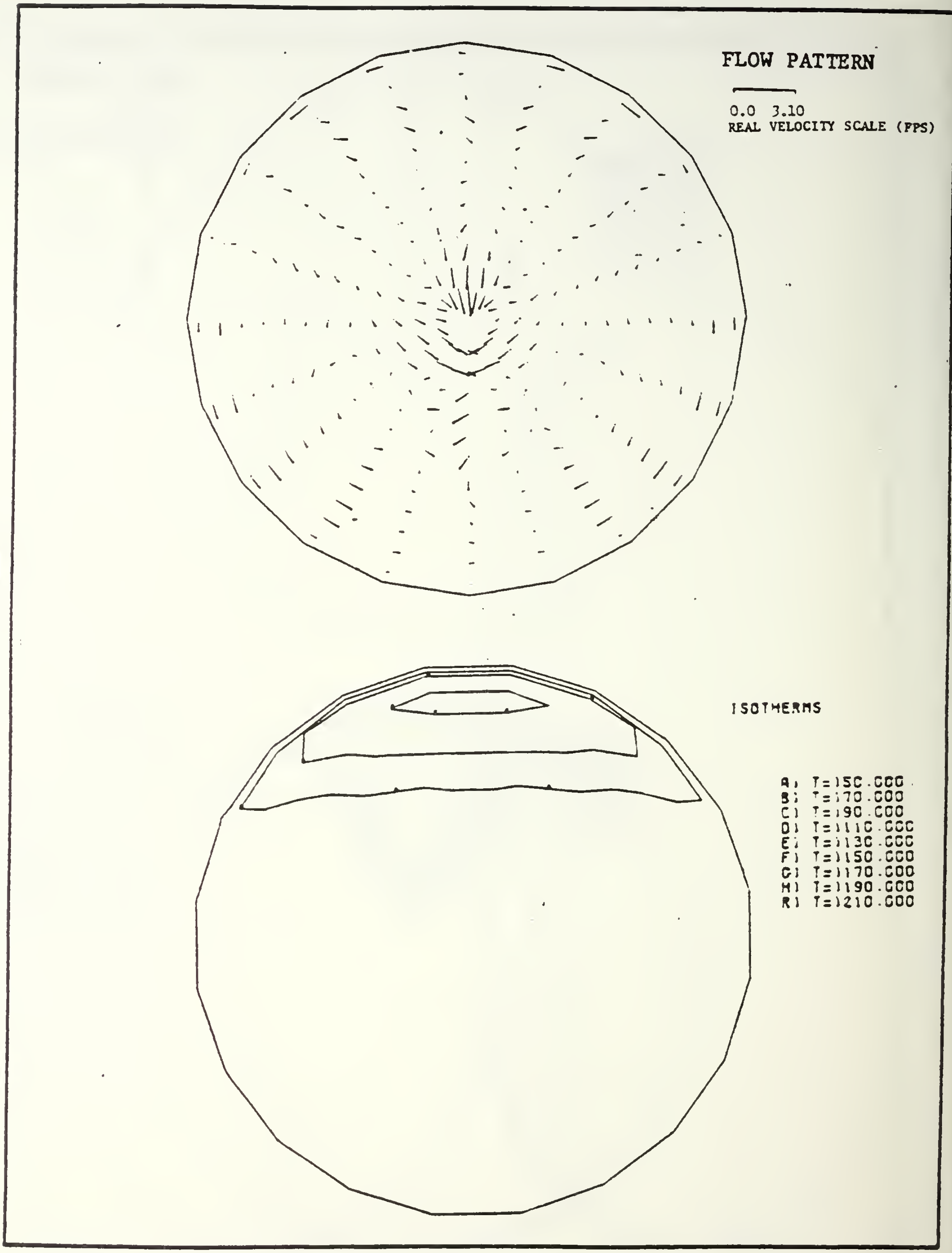


Figure 6.27 Velocity and Isotherm Plots after 90 Sec at the Thermocouple Rack

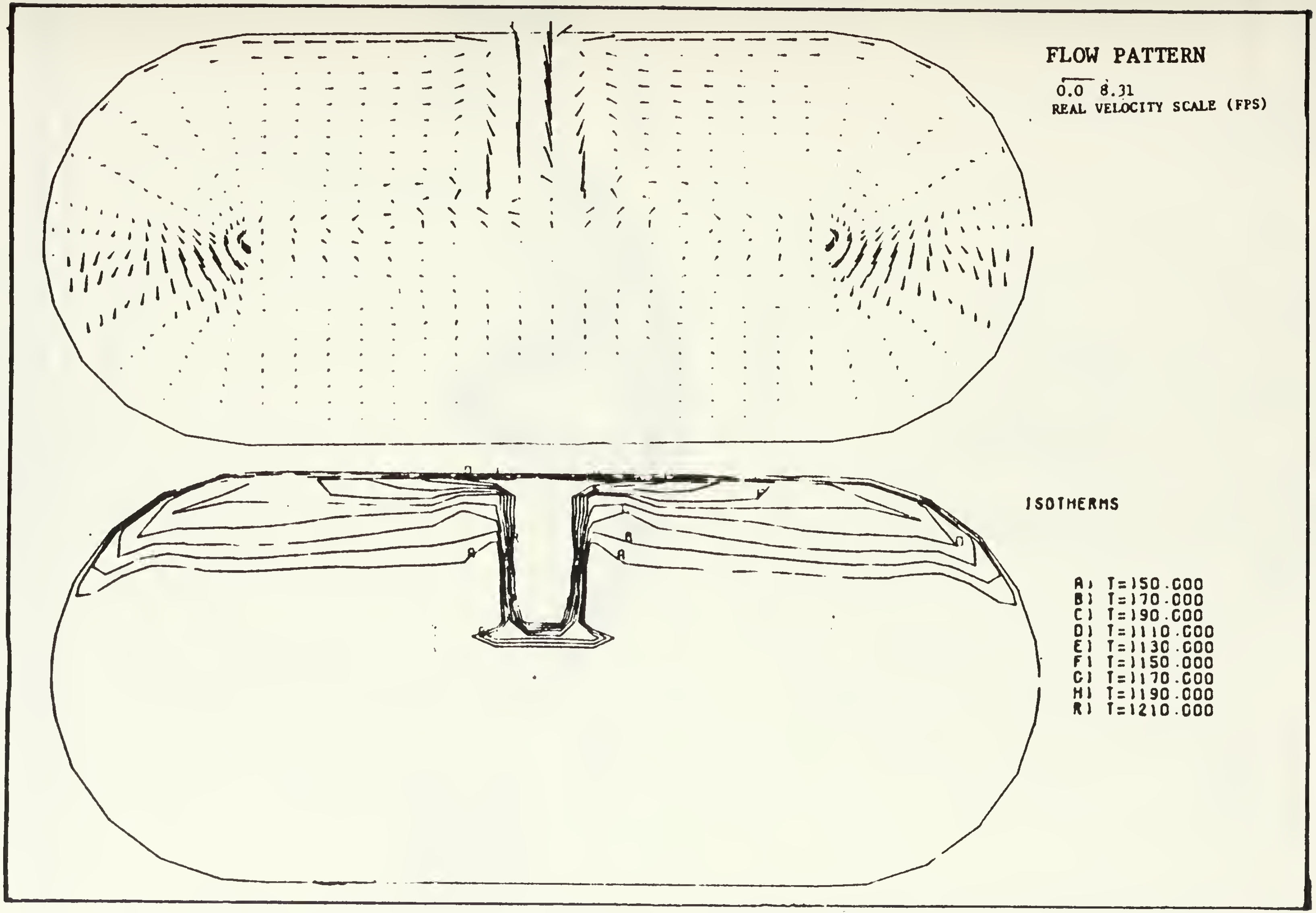


Figure 6.28 Velocity and Isotherm Plots after 130 Sec at Midplane

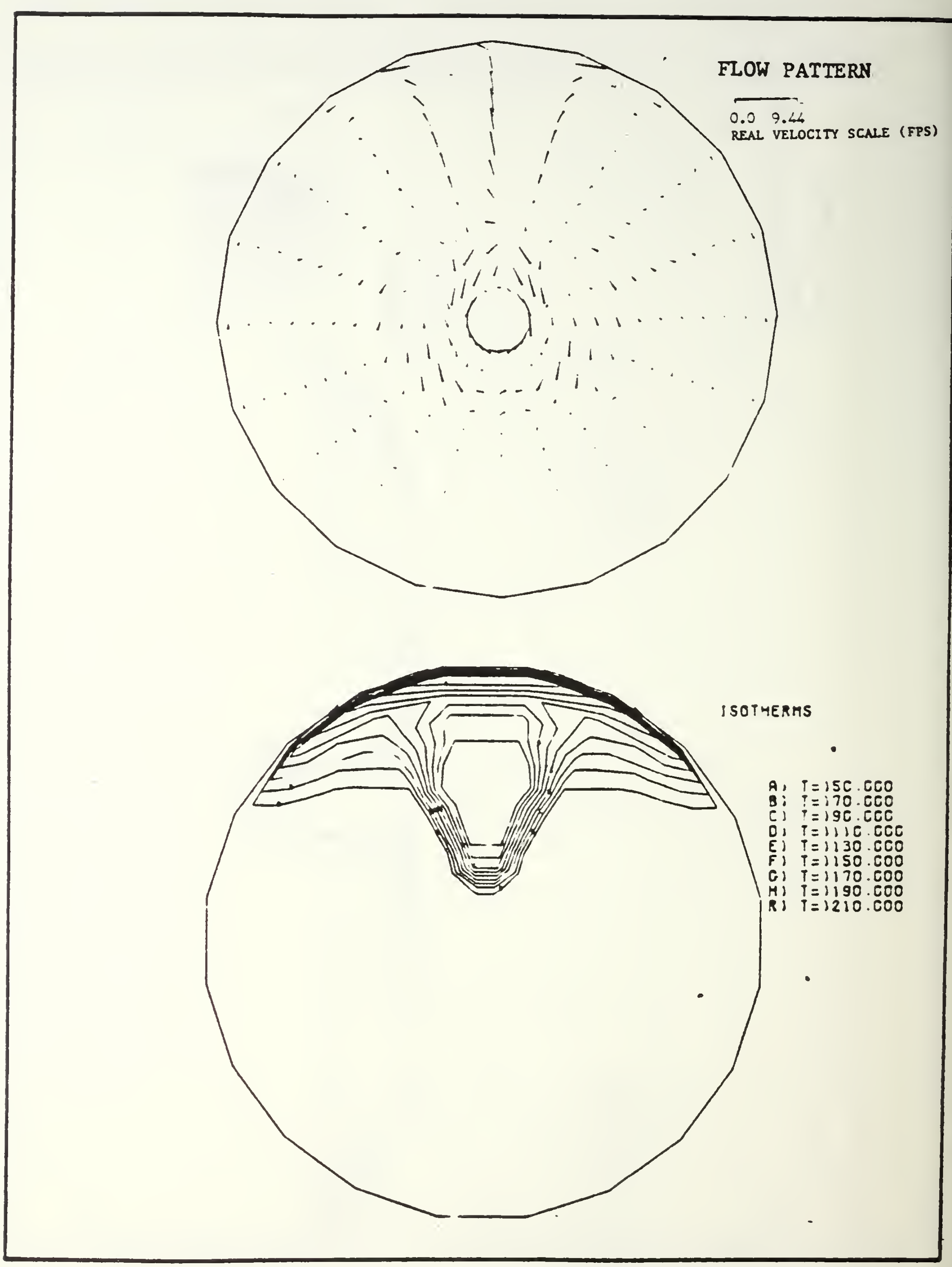


Figure 6.29 Velocity and Isotherm Plots after 130 Sec at Fire Center

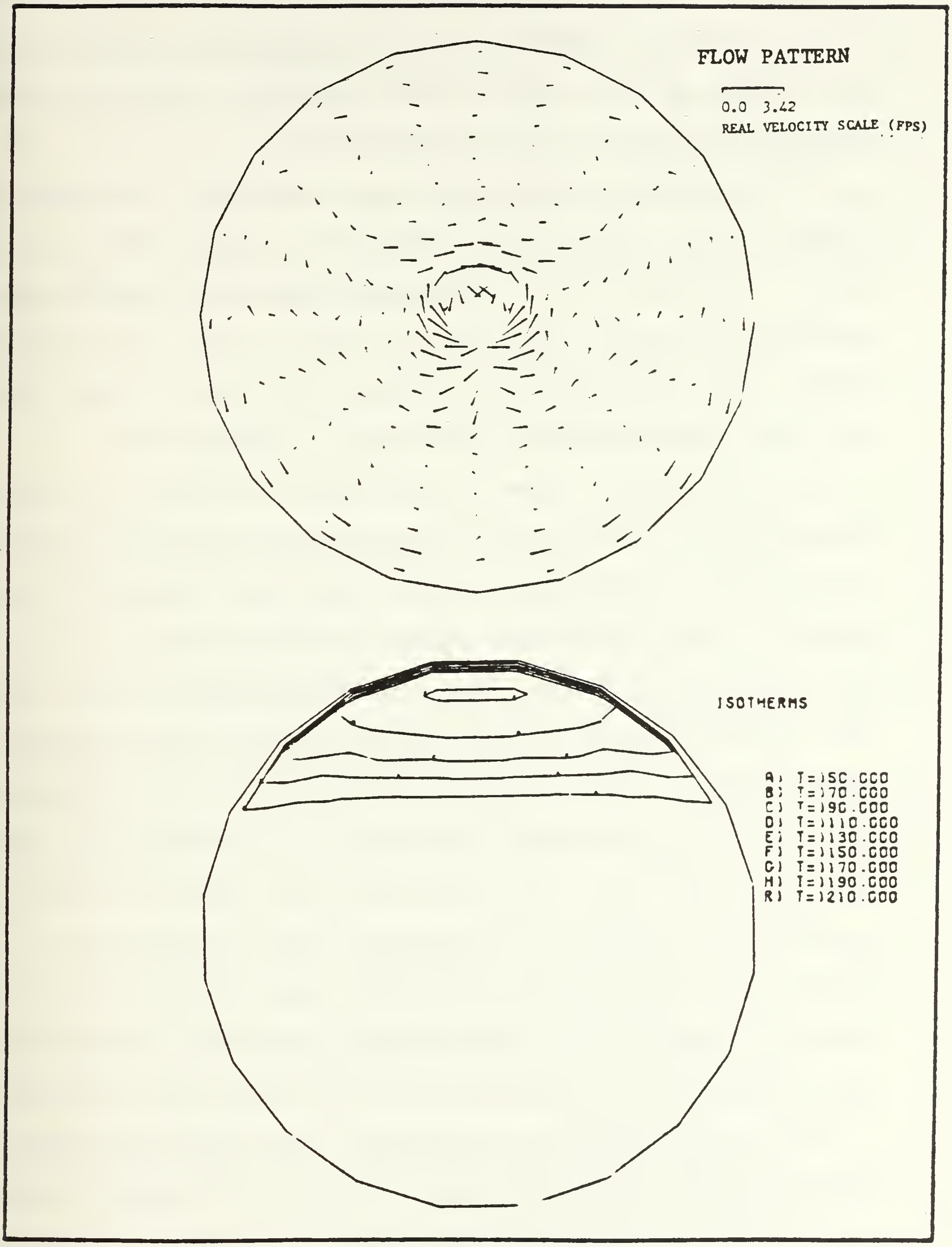


Figure 6.30 Velocity and Isotherm Plots after 130 Sec at the Thermocouple Rack

display the data. These plots can be misleading, especially if the velocity appears to be flowing to a single point. The single point actually represents the tip of another velocity vector in the third dimension.

The velocity fields were not plotted using the same length scale. This can be deceiving unless aware of this point. Therefore some of the plots appear to exhibit a more pronounced velocity field when in fact it was just the scale used. The isotherms are constant for each of the plots, each line representing a 20°C range in temperature.

At 30 seconds, most of the tank is still at ambient temperature. Isotherms are concentrated above the location of the fire, indicating the hot gases are confined to this region. The cross section at the thermocouple rack does not see any noticeable change of temperature at this time. The velocity field at the midplane of the tank exhibits a strong upward flow in the region of the fire. This flow extends to the overhead resulting in a ceiling jet across the top of the tank. The velocity field tends to follow the geometry of the tank, recirculating back to the center. The lower region of the tank exhibits very little motion. The velocity field at the thermocouple rack has a downward flow that extends to a recirculating flow in the third dimension.

At the following time intervals, the isotherms begin to extend further into the tank. This is a valuable tool to see how the hot gases extend into the tank with respect to

time. The velocity fields show unique flow patterns at the three cross sections at each of the time intervals. The recirculation patterns follow the tank geometry, but do develop a changing flow pattern with respect to velocity and direction. This will become important when smoke is entered into the program to see how smoke penetrates the tank and where it becomes concentrated.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The conclusions drawn from this initial simulation of the spherical/cylindrical geometry of the computer code are as follows:

1) The pressure tracking case, Trial 1, provided a numerically generated heat release curve from other available sources. The pressure was forced to follow the experimental curve causing large oscillations in the heat release and temperature data.

2) Trial 2 used a third order polynomial fit of the experimental data provided by NRL. The pressure and temperature did not oscillate greatly, but the values obtained were very high. This indicated the experimental burn rate data was also too high. It was known at the onset that the heat release data could be off by some unknown scaling factor.

3) Of the three test cases examined, Trial 3 was a better representation of the fire in Fire-1. This case combined the heat release rate levels obtained from Trial 1 with a third order polynomial fit variation from Trial 2. The results were a realistic burn rate curve to use as input into the computer code. The temperature readings at the three thermocouples did correspond to the experimental data during the first 130 sec of fire time. The pressure was maintained below the experimental curve.

4) A realistic flow pattern has been observed for Trial 3. The fire plume was shown to increase the velocity of the gas toward the overhead resulting in a ceiling jet. The flow follows the geometry of the tank and develops various recirculating flow patterns with respect to time.

5) The isotherm field plots for Trial 3 illustrate how the hot gases are concentrated in the overhead. With time, the isotherms begin to stratify and penetrate the lower regions of the tank.

6) Trial 3 will be continued to ensure the temperature and pressure curves maintain a proper level in comparison to the experimental data beyond 130 seconds. Further validation of the computer model should be done with accurate heat release rate data before additional complexities are incorporated into the model.

B. RECOMMENDATIONS

The following are recommendations for the future work regarding computer simulation of a fire in Fire-1:

1) Continue with the code validation of the spherical/cylindrical geometry by validating the model with experimental heat release data.

2) Explore the possibility of transferring the program to a supercomputer. The large amount of CPU time it takes to run this program necessitates the use of a larger, faster computer.

3) Incorporate more computer graphics to display the results. This program generates a huge amount of data. The best way to fully understand what is happening in Fire-1 is to see it displayed in three dimensional form. The use of color graphics would be the more preferable option.

4) Begin adding complex interior partitioning. The next step in the computer code would be the addition of decks and recirculating fans. The computer code has been developed and is now waiting for validation studies.

5) Improve the physical models already present and incorporate other models. A combustion model will be added to the computer code to account for the distribution of the heat release rates from the flame. Gaseous radiation will be included in the radiation model and the turbulence model will be updated.

6) The ultimate goal of this project is to develop a computer model that will be able to simulate a shipboard fire scenario. This model will then be able to assist in the design of a ship and fire control tools.

APPENDIX A
FORTRAN LISTING OF THE RADIATION MODEL

```
C***********************************************************************
C          COMPUTER PROGRAM FOR THREE-DIMENSIONAL SURFACE              *
C          RADIATION FOR THE SPHERICAL/CYLINDRICAL GEOMETRY            *
C              OF THE NAVY STORAGE TANK, FIRE-1                        *
C                                                                      *
C                      DEVELOPED BY                                    *
C               J.K. RAYCRAFT AND M.D. KELLEHER                        *
C                                                                      *
C                  NAVAL POSTGRADUATE SCHOOL                           *
C                     MONTEREY, CA 93940                               *
C                                                                      *
C                      AUGUST 1987                                     *
C***********************************************************************
C                                                                      *
C         THERE ARE MI DIVISIONS IN THE THETA DIRECTION                *
C                   MJ DIVISIONS IN THE  R  DIRECTION                  *
C                   MK DIVISIONS IN THE  Z  DIRECTION                  *
C                      MKN ON THE NORTH SPHERICAL END CAP              *
C                      MKS ON THE SOUTH SPHERICAL END CAP              *
C                      MKC ON THE CYLINDRICAL PORTION OF THE TANK      *
C                      ( MKN + MKS + MKC = MK)                         *
C                                                                      *
C        FOR THE NORTH SPHERICAL END CAP THERE ARE MI*MKN DIVISIONS    *
C        NUMBERED   1 TO MI*MKN                                        *
C        FOR THE CYLINDER THERE ARE MI*MKC DIVISIONS                   *
C        NUMBERED  MI*MKN + 1  TO MI * (MKN + MKC)                     *
C        FOR THE SOUTH SPHERICAL END CAP THERE ARE MI*MKS DIVISIONS    *
C        NUMBERED  MI*(MKN + MKC) +1  TO  MI*MK                        *
C                                                                      *
C        MREGN1 REPRESENTS THE NORTH SPHERICAL END CAP                 *
C        IREGN1 IS THE FINAL CELL NUMBER IN THE NORTH CAP (MI*MKN)     *
C        MREGN2 REPRESENTS THE CYLINDER, STARTING AT IREGN1 + 1        *
C        IREGN2 IS THE FINAL CELL NUMBER IN THE CYLINDER (MI*(MKN+MKC)) *
C        MREGN3 REPRESENTS THE SOUTH SPHERICAL END CAP,START AT IREGN2+1*
C        IREGN3 IS THE FINAL CELL NUMBER IN THE SOUTH CAP (MI*MK)      *
C                                                                      *
C        THE SCHEME TO NUMBER THE CELLS IS AS FOLLOWS:                 *
C                   KSM1,KSM2     REPRESENT THE NORTH SPHERE           *
C                                 1 TO IREGN1                          *
C                   KSM3,KSM4     REPRESENT THE CYLINDER               *
C                                 MREGN2 TO IREGN2                     *
C                   KSM5,KSM6     REPRESENT THE SOUTH SPHERE           *
C        THE REASON EACH REGION IS REPRESENTED BY TWO SETS OF THE SAME *
C        NUMBERS IS BECAUSE THE REGIONS "SEE" THEMSELVES, IE. A NORTH  *
C        SPHERE CELL SEES OTHER NORTH SPHERE CELLS AND THERE HAS TO BE *
C        A WAY TO REPRESENT VFMXR(5,5).                                *
C                                                                      *
C        THE CELL NUMBERING IS ACCOMPLISHED IN "DO LOOPS".  STARTING AT *
C        THE NORTH END, THE FIRST VALUE FOR THETA AND K IS CELL ONE,   *
C        THE STARTING NUMBER FOR EACH REGION IS KSM_(MINIMUM VALUE)    *
C        THE VIEW FACTOR TO ALL OTHER CELLS IS ACCOMPLISHED BY FIRST   *
C        FINDING THE OTHER CELLS IN THE SPHERE, KSM2 VARIES FROM 1 TO  *
C        MI*MK, WITH THETA VARYING CCW FIRST THEN Z.  EACH TIME THROUGH *
C        THE DO LOOP TO CHANGE EITHER THETA OR Z, KSM2 INCREASES BY 1. *
C        ONCE THE CYLINDER BOUNDARY IS REACHED, A NEW INNER DO LOOP IS *
C        USED, THIS TIME VARYING KSM3, AND THEN KSM5 FOR THE SOUTH     *
C        SPHERE.                                                       *
C        THE NUMBERING SYSTEM FOR THE CELLS IS FROM NORTH TO SOUTH,    *
C        SPIRALING AROUND THE TANK, VARYING THETA THEN Z. RECIPROSITY  *
C        IS USED TO FIND CORRESPONDING VALUES IN THE MATRIX, IE.       *
C        VFMXR(5,250) AND VFMXR(250,5).                                *
```

```
C                                                                             *
C                                                                             *
C       THE NODE NUMBERING SYSTEM IS AS FOLLOWS:                              *
C             THETA DIRECTION                                                 *
C               STARTING NODE =  NIS ,   FINAL NODE = NI                      *
C               MI = NI - NIS                                                 *
C             Z DIRECTION                                                     *
C               STARTING NODE (NORTH END) = NKS                               *
C               NODE BETWEEN NORTH END CAP AND CYLINDER = NA                  *
C               NODE BETWEEN CYLINDER AND SOUTH END CAP = NB                  *
C               WHERE     MK = NK - NKS                                       *
C                         NA = NKS + MKN                                      *
C                         NB = NKS + ( MKN + MKC)                             *
C                                                                             *
C***************************************************************************
      COMMON/BL1/ NIS,NI,NKS,NK,NA,NB,MI,MK,MKN,MKS,MKC,CL,DTHETA,
     & DPHIN,DPHIS,DZ1,DZ2,DZ3,Z1,R,PI,ZCYL1,ZCYL2
      COMMON/BL2/ PHI(33),THETA(2:21,33),Z(2:21,33),AREA(10),AREAC
      COMMON/BL3/ MREGN1,MREGN2,MREGN3,IREGN1,IREGN2,IREGN3,KSM1,KSM2,
     & KSM3,KSM4,KSM5,KSM6
      COMMON/BL4/ VFMXR(579,579), DELY(2,12),RF
      COMMON/BL5/WVFNN(2:21,3:7,2:21,3:7),WVFSS(2:21,26:30,2:21,26:30),
     & WVFSN(2:21,26:30,2:21,3:7),WVFNC(2:21,3:7,2:21,8:25),
     & WVFCS(2:21,8:25,2:21,26:30), WVFSC(2:21,26:30,2:21,8:25),
     & WVFNS(2:21,3:7,2:21,26:30),WVFCN(2:21,8:25,2:21,3:7),
     &WVFCC(2:21,8:25,2:21,8:25)
      COMMON/BL7/ NJS,NJ,MJ,HSZ,FPAND,HSANG(2,12),Y(2,12),HSY,DIAFP,
     &VFHNS(2,12,2:21,3:7),VFHSS(2,12,2:21,26:30),VFHC(2,12,2:21,8:25),
     &VFNSH(2:21,3:7,2,12),VFSSH(2:21,26:30,2,12),VFCH(2:21,8:25,2,12)
      COMMON/BLK8/VFMXC(579,579),VFMXIN(579,579),
     & CONSRA, NHSZ,AR(579),EM(579),IFIRE
C*******************************************************************************
C        THE MAIN PROGRAM ESTABLISHES THE REQUIRED INPUT VARIABLES FOR THE   *
C        SUCESSFUL RUN OF THE PROGRAM.  IT ALSO WILL CALCULATE THE SIZE       *
C        OF THE REGIONS INVOLVED FOR THE 'I,J,K' INDICIES.                    *
C        FROM THIS PROGRAM ALL OF THE OTHER SUBROUTINES ARE CALLED.           *
C        THE DEFINITION OF THE VARIABLES USED IN THIS SUBROUTINE ARE AS       *
C        FOLLOWS:                                                             *
C              NKS          =  STARTING NODE NUMBER FOR THE K INDICE (Z)      *
C              NA           =  NODE NUMBER BETWEEN THE NORTH SPHERE AND THE   *
C                              CYLINDER                                       *
C              NB           =  NODE NUMBER BETWEED THE CYLINDER AND THE       *
C                              SOUTH SPHERE                                   *
C              NK           =  FINAL NODE NUMBER FOR THE K INDICE (SOUTH END  *
C              NIS          =  STARTING NODE NUMBER FOR THE I INDICE (THETA)  *
C              NI           =  FINAL NODE NUMBER FOR THE I INDICE             *
C              NJS          =  STARTING NODE NUMBER FOR THE J INDICE (R)
C              NJ           =  FINAL NODE NUMBER FOR THE J INDICE
C              R            =  THE RADIUS OF BOTH THE SPHERE AND CYLINDER (FT)*
C              CL           =  THE CYLINDER LENGTH ALONG THE Z AXIS  (FT)     *
C              ZCYL1        =  EVERYTHING IS MEASURED FROM THE NORTH END,     *
C                              THEREFORE THE Z AXIS GOES FROM 0 TO 48.6 (TOTAL*
C                              LENGTH OF THE PRESSURE VESSEL) THIS IS THE Z   *
C                              DISTANCE WHERE THE CYLINDER STARTS     (FT)    *
C              ZCYL2        =  THE DISTANCE WHERE THE CYLINDER STOPS AND THE  *
C                              SOUTH SPHERE BEGINS    (FT)                    *
C              FPAND        =  THE NUMBER OF FIRE CELLS THAT ARE BELOW THE    *
C                              FIRE PAN                                       *
C              DIAFP        =  THE DIAMETER OF THE FIRE PAN (FT)              *
C              HSZ          =  THE Z DISTANCE WHERE THE HEAT SOURCE IS LOCATED*
C              IFIRE        =  A DUMMY VARIABLE THAT WHEN EQUAL TO 0 WILL     *
C                              ALLOW THE PROGRAM TO IGNORE THE SHADING CAUSED *
C                              BY THE FIRE CELLS.  IF THE FIRE IS CONSIDERED  *
C                              IFIRE SHOULD BE ANYTHING EXCEPT ZERO.  SHADING *
C                              REFERS TO THE VIEW FACTORS BETWEEN TANK CELLS  *
C              RF           =  RADIUS OF THE FIRE PAN (FT)                    *
```

```
C
C        GRID   THE SUBROUTINE THAT CALCULATES THE AREAS OF THE CELLS
C               PLUS THE Z AND THETA LOCATIONS FOR EACH CELL.
C
C        WALL   THE SUBROUTINE THAT CALCULATES THE VIEW FACTORS FROM
C               ONE TANK CELL TO ANOTHER.  IF THE FIRE INTERSECTS THE
C               LINE OF SIGHT OF THE CELLS, IT SHADES THAT VIEW FACTOR
C               AND THE VIEW FACTOR IS SET TO ZERO.  THE MATRICES HAVE
C               FOUR INDICIES.  I,K  THETA AND Z LOCATIONS FROM THE
C               STARTING CELL. II,KK THETA AND Z LOCATIONS TO THE CELL
C               THE RADIATON IS GOING TO. NOTE BE CAREFUL OF THE INDICE
C               NOTATION IN THE COMMON STATEMENTS.  #1:#2 IN THE
C               NOTATION MEANS #1 = CELL STARTING FROM  #2 = CELL GOING
C               THROUGH.  FOR EXAMPLE WVFSN(2:21,26:30,2:21,3:7)
C               WVFSN = WALL VIEW FACTOR SOUTH SPHERE TO NORTH SPHERE
C               (I,K,II,KK) FOR BOTH TIMES WHERE 2:21 APPEARS IT STANDS
C               FOR I AND II MEANING THE THETA CELLS WHICH GO THROUGH
C               CELL NUMBER 2 THROUGH CELL NUMBER 21.  THESE ARE CELL
C               NUMBERS NOT NODE NUMBERS.  THE SOUTH SPHERE HAS CELLS
C               26 THROUGH 31 AND THE NORTH SPHERE HAS CELLS 3 THROUGH
C               7 WHICH CORRESPOND TO K AND KK RESPECTFULLY.  IF THIS
C               PROGRAM IS TO BE MODIFIED IN THE FUTURE THE COMMON
C               STATEMENTS WILL HAVE TO BE CHANGED TO CORRESPOND TO
C               THE RIGHT CELL NUMBERS.  THIS IS ONLY IF THE PROGRAM IS
C               TO BE ENLARGED.  A SMALLER MATRIX CAN BE RUN WITH THIS
C               PROGRAM BUT NOT ALL THE SPACE SET ASIDE WOULD BE USED.
C               THE INTERNAL PROGRAM IS GENERAL AND WILL NOT HAVE TO
C               BE CHANGED.
C
C        VIEW   THIS SUBROUTINE TAKES THE WALL VIEW FACTORS AND PUTS
C               THEM INTO ONE ARRAY WITH TWO INDICES VICE FOUR.  IT
C               ASSIGNS A NUMBER TO EACH OF THE CELLS FROM 1 TO 560.
C               THE NUMBERING IS FROM THE NORTH ENDCAP AND SPIRALS
C               AROUND THE TANK.  THETA VARIES THEN Z.
C
C        HEAT  THIS SUBROUTINE INTRODUCES THE FIRE CELLS INTO THE
C              VIEW FACTOR MATRIX.  FIRST THE VIEW FACTORS FROM THE
C              FIRE TO EACH TANK CELL ARE FOUND.  THESE VIEW FACTORS
C              ARE THEN MODIFIED DUE TO THE PROBLEMS EXPERIENCED WITH
C              GEOMETRY OF THE TANK.  THE VIEW FACTORS FROM THE WALL TO
C              THE FIRE ARE FOUND THROUGH RECIPROSITY, AND MODIFIED
C              DUE TO THE UNKNOWN EXACT AREA OF THE FIRE. THE FOUR
C              INDICE MATRIXS ARE THEN PUT INTO THE PREVIOUS TWO INDICE
C              MATRIX.  THE FIRE CELLS ARE NUMBERED FROM THE FIRE PAN
C              TO THE TOP OF THE TANK.(CELLS 561 - 579) THE VIEW
C              FACTORS FROM THE FIRE CELLS TO THE FIRE CELLS ARE ALL
C              SET TO ZERO.
C
C        AREA1 THIS SUBROUTINE ASSIGNS AN AREA TO EACH CELL. 'GRID'SET
C              UP A GENERAL AREA THAT COULD BE USED IN 'WALL' BUT IN
C              ORDER TO USE THE 'INVER' SUBROUTINE AN EASIER WAY HAD
C              TO BE DEVELOPED TO INDICATE AREA.
C
C        INVER THIS SUBROUTINE TAKES THE VIEW FACTOR MATRIX, INCLUDING
C              THE FIRE CELLS AND SETS IT IN THE EQUATION DEVELOPED IN
C              SPEIGAL/HOWELL TO FIND THE HEAT TRANSFER RATE.  THE GOAL
C              OF THIS PROGRAM INVOLVES MODIFIYING THE VIEW FACTOR
C              MATRIX AND TAKING THE INVERSE TO PROVIDE A NEW MATRIX
C              THAT WILL BE USED IN THE "TANK" PROGRAM TO CALCULATE
C              HEAT TRANSFER.  THIS NEW MATRIX IS REFERED TO AS VFMXC
C              IN THE PROGRAM OR TO THE "G" MATRIX FOR DISCUSSION
C              PURPOSES.  DO TO THE NUMBER OF LARGE MATRICES REQUIRED
C              FOR THIS MODIFICATION/INVERSE PROCEDURE, A SPACE SAVING
C              PROCEDURE WAS USED TO WRITE OVER MATRIX LOCATIONS.
C              THE IBM INSL PROCEDURE, LINV1F, WAS CALLED TO DO THE
C              MATRIX INVERSION.  AFTER THE MATRIX IS INVERTED IT IS
C              MULTIPLIED BY ANOTHER MATRIX AND THE RESULTING "G"
C              MATRIX IS SENT TO A DISK FOR USE IN THE TANK PROGRAM.
C
C    THIS IS A GENERAL OVERVIEW WHAT THE PROGRAM DOES.  EACH SUBROUTINE
```

```
C        WILL HAVE INTERNAL COMMENTS AND A BEGINNING SECTION TO DEFINE THE   *
C        VARIABLES USED.                                                      *
******************************************************************************
C     READ(5,*) NKS,NA,NB,NK,NIS,NI,R,CL,ZCYL1,ZCYL2,FPAND,NJS,NJ,HSZ,
C    & IFIRE
      PI = 4.0 * ATAN(1.0)
      NKS = 3
      NA  = 8
      NB  = 26
      NK  = 31
      NIS = 2
      NI  = 22
      R   = 9.6
      CL  = 27.4
      ZCYL1 = 9.6
      ZCYL2 = 37.0
      FPAND = 5
      DIAFP = 2.
      NJS = 1
      NJ = 13.
      HSZ = 23.3
      IFIRE = 1
      RF = DIAFP/2.
      MI = NI - NIS
      MJ = NJ - NJS
      MK = NK - NKS
      MKN = NA - NKS
      MKS = NK - NB
      MKC = MK - MKN - MKS
      WRITE(6,*) 'NKS =', NKS, 'NA =',NA,'NB =',NB,'NK =',NK
      WRITE(6,*) 'NIS =', NIS, 'NI =',NI, 'NJS =',NJS,'NJ =',NJ
      WRITE(6,*)
      WRITE(6,*) 'MI =',MI,'MK =',MK,'MKN =',MKN,'MKS =',MKS
      WRITE(6,*) 'MKC =',MKC, 'MJ =',MJ
      WRITE(6,*)
      CALL GRID
      CALL WALL
      MREGN1 = 0
      IREGN1 = MI * MKN
      MREGN2 = IREGN1 +1
      IREGN2 = MI * (MKN + MKC)
      MREGN3 = IREGN2 + 1
      IREGN3 = MI * MK
      CALL VIEW
      CALL HEAT
C  THE FOLLOWING DO LOOP ADDS UP THE VIEW FACTORS FROM I TO ALL THE
C  OTHER 579 CELLS.  THIS IS A CHECK OF THE ENCLOSURE PROPERTY FOR THE
C  TOTAL SUM SHOULD EQUAL ONE.
      DO 46 I = 1,579
         SUM = 0.0
      DO 47 J = 1,579
         SUM = SUM + VFMXR(I,J)
   47 CONTINUE
      WRITE(6,*) I, 'SUM TOTAL =  ' , SUM
   46 CONTINUE
      CALL AREA1
      CALL INVER
      STOP
      END


      SUBROUTINE GRID
      COMMON/BL1/ NIS,NI,NKS,NK,NA,NB,MI,MK,MKN,MKS,MKC,CL,DTHETA,
     & DPHIN,DPHIS,DZ1,DZ2,DZ3,Z1,R,PI,ZCYL1,ZCYL2
      COMMON/BL2/ PHI(33),THETA(2:21,33),Z(2:21,33),AREA(10),AREAC
```

```
      COMMON/BL3/ MREGN1,MREGN2,MREGN3,IREGN1,IREGN2,IREGN3,KSM1,KSM2,
     & KSM3,KSM4,KSM5,KSM6
      COMMON/BL4/ VFMXR(579,579),DELY(2,12),RF
      COMMON/BL5/WVFNN(2:21,3:7,2:21,3:7),WVFSS(2:21,26:30,2:21,26:30),
     & WVFSN(2:21,26:30,2:21,3:7),WVFNC(2:21,3:7,2:21,8:25),
     & WVFCS(2:21,8:25,2:21,26:30), WVFSC(2:21,26:30,2:21,8:25),
     & WVFNS(2:21,3:7,2:21,26:30),WVFCN(2:21,8:25,2:21,3:7),
     &WVFCC(2:21,8:25,2:21,8:25)
      COMMON/BL7/ NJS,NJ,MJ,HSZ,FPAND,HSANG(2,12),Y(2,12),HSY,DIAFP,
     &VFHNS(2,12,2:21,3:7),VFHSS(2,12,2:21,26:30),VFHC(2,12,2:21,8:25),
     &VFNSH(2:21,3:7,2,12),VFSSH(2:21,26:30,2,12),VFCH(2:21,8:25,2,12)
      COMMON/BLK8/VFMXC(579,579),VFMXIN(579,579),
     & CONSRA, NHSZ,AR(579),EM(579),IFIRE
************************************************************************
*       DTHETA        =    DELTA THETA (I DIRECTION)                   *
*       DPHIN         =    DELTA PHI NORTH SPHERE                      *
*       DPHIS         =    DELTA PHI SOUTH SPHERE                      *
*       Z(I,K)        =    Z ARRAY TO ASSIGN A Z LOCATION FOR EVERY CELL*
*       THETA(I,K)    =    THETA ARRAY TO ASSIGN A THETA LOCATION      *
*       PHI(I)        =    ASSIGNS A PHI VALUE FOR THE NODAL POINT ON  *
*                          THE SPHERES VICE THE CELL POINT             *
*       AREA(I)       =    AN AREA ELEMENT FOR THE CELLS ON THE SPHERE *
*                          NOTE FOR A GIVEN Z LOCATION ALL THE CELLS HAD*
*                          THE SAME AREA.  THEREFORE ONLY 10 LOCATION  *
*                          SITES HAD TO BE ASSIGNED.                   *
*       AREAC         =    THE AREA OF THE CYLINDER CELLS.  DUE TO THE *
*                          UNIFORM GRID ON THE TANK WALLS, THE CELLS   *
*                          THE SAME SIZE.                              *
*       ANGLE         =    USED TO ASSURE THAT THE ANGLE AT NK WAS 180 *
*                          DEGREES AND THEREFORE THE COS(180) = -1.    *
*                          THERE WAS SOME PROBLEM DETECTED EARLY IN    *
*                          TESTING THAT WOULD MAKE THIS VALUE POSITIVE *
*                          DUE TO COMPUTER ROUND OFF                   *
************************************************************************
C     DEFINE THE GRID SYSTEM
      DTHETA = ( 2.0 * PI) / MI
      WRITE(6,*) 'DTHETA = ', DTHETA
C     FOR THE SPHERICAL END CAPS, PHI IS 90 DEGREES OR PI/2 DIVIDED BY
C     THE NUMBER OF DIVISIONS PER END CAP
      DPHIN = PI / (2.0 * MKN)
      DPHIS = PI / (2.0 * MKS)
      WRITE(6,*) 'DPHIN =',DPHIN,'DPHIS =',DPHIS
      WRITE(6,*)
C     PHI IS FOUND FOR EACH NODAL POINT.  THIS IS NOT THE PHI FOR THE
C     MIDPOINT OF THE CELL.
C     PHI FOR THE NORTH SPHERE IS FROM  0 TO PI/2  RADIANS
      PHI(NKS) = 0.0
      WRITE(6,50) 'I', 'PHI (RADIANS)'
 50   FORMAT(1X, 10X,A,10X,A,/)
      DO 1 I = NKS+1, NA
          PHI(I) = PHI(I-1) + DPHIN
      WRITE(6,55) I, PHI(I)
  1   CONTINUE
C     PHI FOR THE SOUTH SPHERE IS FROM PI/2 TO PI RADIANS
      PHI(NB) = PI / 2.0
      DO 2 I = NB+1, NK
          PHI(I) = PHI(I-1) + DPHIS
      WRITE(6,55) I, PHI(I)
  2   CONTINUE
 55   FORMAT(1X,7X,I3,10X,F10.5)
C     DEFINE THE LOCATION OF EACH CELL IN TERMS OF THETA AND Z, SET UP
C     A MATRIX FOR EACH.   THE LOCATION IS IN THE MIDDLE OF EACH CELL.
C     THE CELL AREAS ARE THE SAME FOR EACH ELEMENT ON THE CYLINDER AND
C     SIMILAR FOR EACH PHI ANGLE OF THE SPHERE. SET UP A COLUMN VECTOR
```

```
C     FOR THE AREA OF EACH CELL TYPE.
C     START AT THE NORTH SPHERE WITH Z1 = 0.  FIND DELTAZ. ( WILL VARY
C     FOR THE SPHERE).  TAKE THE MIDPOINT, THIS IS THE LOCATION FOR THE
C     FIRST CELL. ADD THE REMAINING DELTAZ (DZ/2).  THIS LOCATION IS NOW
C     AT THE NEXT NODAL POINT. GO THROUGH THE LOOP AGAIN TO GET A NEW
C     DELTAZ AND CONTINUE WITH THE PROCESS.  NOTE THE NODAL POINT NUMBER
C     REPRESENTS THE CELL TO THE RIGHT OF IT GOING FROM NORTH TO SOUTH.
C     THIS IS FOR THE Z DIRECTION.  THE LAST CELL HAS THE NUMBER NK-1.
C     NA REPRESENTS A CELL ON THE CYLINDER, NB-CELL ON THE SOUTH SPHERE
       Z1 = 0.0
       KT = 0
C     WRITE(6,60)'I','K','THETA','Z'
       DO 3 K = NKS,NK-1
       DO 3 I = NIS, NI-1
           IF ( KT .EQ. K) THEN
                 Z(I,K) = Z(I-1,K)
                 GO TO 75
           ENDIF
           KT = K
C  FIND THE AREA AND Z LOCATION FOR THE NORTH SPHERE
           IF ( K .LT. NA ) THEN
                 DZ1 = R * ( COS(PHI(K)) - COS(PHI(K+1)) )
                 AREA(K-NKS+1) = (DZ1* 2.*PI*R)/ MI
                 Z(I,K) = Z1 + (DZ1*0.5)
                 Z1 = Z(I,K) + (DZ1*0.5)
C FIND THE Z LOCATION FOR THE CYLINDER
           ELSE IF ( K .GE. NA .AND. K .LT. NB) THEN
                 DZ2 = CL / MKC
                 Z(I,K) = Z1 + (DZ2*0.5)
                 Z1 = Z(I,K) + (DZ2*0.5)
           ELSE
C FIND THE AREA AND Z LOCATION FOR THE SOUTH SPHERE
                 IF (K .EQ. NK-1 ) THEN
C ENSURES THAT THE ANGLE AT ((NK-1) +1),IE NK, EQUALS 180 DEGREES
                       ANGLE = -1.0
                 ELSE
                 ANGLE = COS(PHI(K+1))
                 ENDIF
                 DZ3 = R*( COS(PHI(K)) - ANGLE)
                 AREA(K+MKN+1-NB) = (DZ3*2.0*PI*R)/ MI
                 Z(I,K) = Z1 + (DZ3*0.5)
                 Z1 = Z(I,K) + (DZ3*0.5)
           ENDIF
  75  CONTINUE
C ASSIGNS A THETA VALUE FOR EVERY CELL.
           THETA(I,K) = (I-NIS+1)*DTHETA - (0.5 * DTHETA)
C             WRITE(6,65) I,K,THETA(I,K),Z(I,K)
  3   CONTINUE
  60  FORMAT(1X,5X,A,5X,A,7X,A,10X,A,/)
  65  FORMAT(1X,2X,I3,2X,I3,4X,F10.5,6X,F10.5)
C     THE AREA VECTOR IS NUMBERED 1-MKN FOR THE NORTH SPHERE AND MKN+1
C     TO MKN+MKS FOR THE SOUTH SPHERE.  THE AREA FOR THE CYLINDER CELL
C     IS CONSTANT AND CAN BE CALCULATED OUTSIDE THE DO LOOP.
       AREAC = R*DZ2*DTHETA
C     WRITE(6,*) 'AREAC = ', AREAC
C     DO 70 I = 1,10
C        WRITE(6,*) I, AREA(I)
C 70  CONTINUE
       RETURN
       END
       SUBROUTINE WALL
       COMMON/BL1/ NIS,NI,NKS,NK,NA,NB,MI,MK,MKN,MKS,MKC,CL,DTHETA,
     & DPHIN,DPHIS,DZ1,DZ2,DZ3,Z1,R,PI,ZCYL1,ZCYL2
       COMMON/BL2/ PHI(33),THETA(2:21,33),Z(2:21,33),AREA(10),AREAC
       COMMON/BL3/ MREGN1,MREGN2,MREGN3,IREGN1,IREGN2,IREGN3,KSM1,KSM2,
```

```
      & KSM3,KSM4,KSM5,KSM6
       COMMON/BL4/ VFMXR(579,579),DELY(2,12),RF
       COMMON/BL5/WVFNN(2:21,3:7,2:21,3:7),WVFSS(2:21,26:30,2:21,26:30),
      & WVFSN(2:21,26:30,2:21,3:7),WVFNC(2:21,3:7,2:21,8:25),
      & WVFCS(2:21,8:25,2:21,26:30), WVFSC(2:21,26:30,2:21,8:25),
      & WVFNS(2:21,3:7,2:21,26:30),WVFCN(2:21,8:25,2:21,3:7),
      &WVFCC(2:21,8:25,2:21,8:25)
       COMMON/BL7/ NJS,NJ,MJ,HSZ,FPAND,HSANG(2,12),Y(2,12),HSY,DIAFP,
      &VFHNS(2,12,2:21,3:7),VFHSS(2,12,2:21,26:30),VFHC(2,12,2:21,8:25),
      &VFNSH(2:21,3:7,2,12),VFSSH(2:21,26:30,2,12),VFCH(2:21,8:25,2,12)
       COMMON/BLK8/VFMXC(579,579),VFMXIN(579,579),
      & CONSRA, NHSZ,AR(579),EM(579),IFIRE
************************************************************************
*      WVFNN     =    WALL VIEW FACTOR NORTH SPHERE TO NORTH SPHERE      *
*      WVFNS     =    WALL VIEW FACTOR NORTH SPHERE TO SOUTH SPHERE      *
*      WVFSN     =    WALL VIEW FACTOR SOUTH SPHERE TO NORTH SPHERE      *
*      WVFSS     =    WALL VIEW FACTOR SOUTH SPHERE TO SOUTH SPHERE      *
*      WVFNC     =    WALL VIEW FACTOR NORTH SPHERE TO CYLINDER          *
*      WVFCN     =    WALL VIEW FACTOR CYLINDER TO NORTH SPHERE          *
*      WVFSC     =    WALL VIEW FACTOR SOUTH SPHERE TO CYLINDER          *
*      WVFCS     =    WALL VIEW FACTOR CYLINDER TO SOUTH SPHERE          *
*      WVFCC     =    WALL VIEW FACTOR CYLINDER TO CYLINDER              *
*                                                                        *
*      PHI1      =    PHI ANGLE TO MIDPOINT OF THE CELL ON A SPHERICAL   *
*                     ELEMENT, ONE DENOTES ORIGINATING CELL              *
*      PHI2      =    PHI ANGLE OF THE CELL THE RADIATION IS GOING TO    *
*      RHO1      =    PROJECTED DISTANCE ON THE XY PLANE (R*SIN(PHI1) )  *
*      RHO2      =    PROJECTED DISTANCE OF THE CELL RADIATION IS GOING TO*
*      H1,H2     =    DISTANCE ALONG THE Z AXIS OF THE SPHERICAL CELLS   *
*                     R * COS(PHI1) 1=ORIGINATING , 2 = RECEIVING        *
*      THETAD    =    THE DIFFERENCE BETWEEN THE THETA ANGLES OF THE     *
*                     TWO CELLS IN QUESTION                              *
*      DAREA1    =    THE AREA OF THE ORIGINATING CELL                   *
*      DAREA2    =    THE AREA OF THE RECEIVING CELL                     *
*      ASQ       =    "A" SQUARE, THIS IS A SQUARED DISTANCE OBTAINED BY *
*                     THE LAW OF COSINES.  THIS DISTANCE IS REQUIRED TO  *
*                     FIND THE DISTANCE BETWEEN THE TWO CELLS.  REFER TO *
*                     THESIS TEXT FIGURES TO UNDERSTAND THE DERIVATIONS  *
*      BSQ       =    "B" SQUARE, AGAIN ANOTHER DISTANCE REQUIRED TO FIND *
*                     THE DISTANCE BETWEEN THE TWO CELLS                 *
*      RSQ       =    "R" SQUARE,  THE SQUARE OF THE DISTANCE BETWEEN THE *
*                     TWO CELLS                                          *
*      RD        =    THE ACTUAL DISTANCE BETWEEN THE TWO CELLS        *
*      CBETA1    =    THE COSINE OF THE ANGLE BETWEEN THE NORMAL OF THE  *
*                     ORIGINATING CELL AND THE LINE RD                   *
*      CBETA2    =    THE COSINE OF THE ANGLE BEWTEEN THE NORMAL OF THE  *
*                     RECEIVING CELL AND THE LINE RD                     *
*      ZETA1S    =    ZETA ONE SQUARE, ANOTHER DISTANCE REQUIRED TO FIND *
*                     CBETA1, REFER TO THESIS TEXT                       *
*      ZETA2S    =    ZETA TWO SQUARE, USED TO FIND CBETA2               *
*                                                                        *
*      THE FOLLOWING VARIABLES ARE USED TO DETERMINE IF THE LINE BETWEEN *
*      THE CELLS INTERSECTS THE FIRE                                     *
*                                                                        *
*      XI,YI,ZI =     THE X,Y, AND Z LOCATION OF THE ORIGINATING CELL    *
*      XJ,YJ,ZJ =     THE X,Y, AND Z LOCATION OF THE RECEIVING CELL      *
*      XD        =    THE X DISTANCE BETWEEN THE TWO (XJ-XI)             *
*      YD        =    THE Y DISTANCE BETWEEN THE TWO (YJ-YI)             *
*      ZD        =    THE Z DISTANCE BETWEEN THE TWO (ZJ-ZI)             *
*      A,B,C     =    COEFICIENTS OF THE EQUATION:                       *
*                     A*T**2 + B*T + C = 0. THE DETERMINATION OF THESE   *
*                     COEFICIENTS IS DISCUSSED IN THE THESIS             *
*      QUAD      =    THE TERMS IN A QUADRATIC SOLUTION THAT WOULD BE    *
*                     UNDER THE SQUARE ROOT SIGN                         *
*      T1,T2     =    SOLUTIONS TO THE QUADRATIC SOLUTION                *
*      Y1,Y2     =    Y DISTANCES THAT CAN BE RELATED TO THE LOCATION OF *
*                     THE FIRE                                           *
*      FIREY     =    THE NEGATIVE Y DISTANCE THAT REPRESENTS WHERE THE  *
```

```
*                      THE FIRE PAN IS LOCATED.                                *
********************************************************************************
C NOTE ORIGINATING CELL IS (I,K), RECEIVING CELL IS (II,KK)
C*******************************************************************************
C     WVFNN  WALL VIEW FACTOR FROM NORTH SPHERE TO NORTH SPHERE
      DO 100 I = NIS,NI-1
      DO 100 K = NKS,NA-1
      DO 100 II = NIS, NI-1
      DO 100 KK = NKS,NA-1
      IF (I .EQ. II .AND. K .EQ. KK) THEN
         WVFNN(I,K,II,KK) = 0.0
      ELSE
         PHI1 = PHI(K) + .5*DPHIN
         PHI2 = PHI(KK) + .5*DPHIN
         RHO1 = R * SIN (PHI1)
         RHO2 = R * SIN (PHI2)
         H1 = R * COS(PHI1)
         H2 = R * COS(PHI2)
         THETAD = ABS( THETA(I,K) - THETA(II,KK))
         IF (THETAD.GT. PI) THETAD = 2*PI - THETAD
         DAREA1 = AREA(K-NKS+1)
         DAREA2 = AREA(KK-NKS+1)
         ASQ = RHO1**2 + RHO2**2 - 2.0*RHO1*RHO2*COS(THETAD)
         BSQ = (H1-H2)**2
         RSQ = ASQ + BSQ
         RD  = SQRT(RSQ)
         CBETA1 = RD/(2.0*R)
         CBETA2 = RD/(2.0*R)
         WVFNN(I,K,II,KK) = (CBETA1*CBETA2)/(PI*RSQ) * DAREA2
      ENDIF
  100 CONTINUE
         WRITE(6,*)
C*******************************************************************************
C     WVFSS  WALL VIEW FACTOR FROM SOUTH SPHERE TO SOUTH SPHERE
      DO 200 I = NIS,NI-1
      DO 200 K = NB,NK-1
      DO 200 II = NIS, NI-1
      DO 200 KK = NB,NK-1
      IF (I .EQ. II .AND. K .EQ. KK) THEN
         WVFSS(I,K,II,KK) = 0.0
      ELSE
         PHI1 = PI - (PHI(K) + .5*DPHIS)
         PHI2 = PI - (PHI(KK) + .5*DPHIS)
         RHO1 = R * SIN (PHI1)
         RHO2 = R * SIN (PHI2)
         H1 = R * COS(PHI1)
         H2 = R * COS(PHI2)
         THETAD = ABS( THETA(I,K) - THETA(II,KK))
         IF (THETAD.GT. PI) THETAD = 2*PI - THETAD
         DAREA1 = AREA(K+MKN+1-NB)
         DAREA2 = AREA(KK+MKN+1-NB)
         ASQ = RHO1**2 + RHO2**2 - 2.0*RHO1*RHO2*COS(THETAD)
         BSQ = (H1-H2)**2
         RSQ = ASQ + BSQ
         RD  = SQRT(RSQ)
         CBETA1 = RD/(2.0*R)
         CBETA2 = RD/(2.0*R)
         WVFSS(I,K,II,KK) = (CBETA1*CBETA2)/(PI*RSQ) * DAREA2
      ENDIF
  200 CONTINUE
         WRITE(6,*)
C*******************************************************************************
C     WVFNS  WALL VIEW FACTOR FROM NORTH SPHERE TO SOUTH SPHERE
      DO 300 I = NIS,NI-1
      DO 300 K = NKS,NA-1
      DO 300 II = NIS, NI-1
```

```
      DO 300 KK = NB,NK-1
      IF (I .EQ. II .AND. K .EQ. KK) THEN
         WVFNS(I,K,II,KK) = 0.0
      ELSE
         PHI1 = PHI(K) + .5*DPHIN
         PHI2 = PI - (PHI(KK) + .5*DPHIS)
         RHO1 = R * SIN (PHI1)
         RHO2 = R * SIN (PHI2)
         H1 = R * COS(PHI1)
         H2 = R * COS(PHI2)
         THETAD = ABS( THETA(I,K) - THETA(II,KK))
         IF (THETAD.GT. PI) THETAD = 2*PI - THETAD
         DAREA1 = AREA(K-NKS+1)
         DAREA2 = AREA(KK+MKN+1-NB)
         ASQ = RHO1**2 + RHO2**2 - 2.0*RHO1*RHO2*COS(THETAD)
         BSQ = (CL + H1 + H2)**2
         RSQ = ASQ + BSQ
         RD  = SQRT(RSQ)
         ZETA1S = RHO2**2 + (CL + H2)**2
         ZETA2S = RHO1**2 + (CL + H1)**2
         CBETA1 = (R**2 + RSQ - ZETA1S)/ (2.0*R*RD)
         CBETA2 = (R**2 + RSQ - ZETA2S)/ (2.0*R*RD)
         WVFNS(I,K,II,KK) = (CBETA1*CBETA2)/(PI*RSQ) * DAREA2
      ENDIF
C*************************************************************************
C THE FOLLOWING SECTION IS ONLY INCLUDED IF THE FIRE IS CONSIDERED.
C IN THAT CASE,IFIRE (INCLUDED FIRE) WOULD EQUAL 1 AND THE CHECK TO
C SEE IF THE VIEW FACTOR INTERSECTS THE FIRE WOULD BE ACCOMPLISHED.
C*************************************************************************

      IF (IFIRE .EQ. 0) GO TO 350
      IF(WVFNS(I,K,II,KK) .EQ. 0.) GO TO 350
      FIREY = -R + (R/MJ * FPAND)
      RF = DIAFP / 2.0
      XI = RHO1*COS(THETA(I,K))
      YI = RHO1*SIN(THETA(I,K))
      ZI = Z(I,K)
      XJ = RHO1*COS(THETA(II,KK))
      YJ = RHO1*SIN(THETA(II,KK))
      ZJ = Z(II,KK)
      XD = XJ - XI
      YD = YJ - YI
      ZD = ZJ - ZI
      A  = XD**2 + ZD**2
      B  = 2.0 *(ZI - HSZ)*ZD + 2.0*XI*XD
      C  = XI**2 + (ZI - HSZ)**2 - RF**2
      QUAD = B**2 - 4.*A*C
         IF ( QUAD .LT. 0.) THEN
               GO TO 350
         ELSE
               T1 = (-B + SQRT(QUAD))/(2.*A)
               T2 = (-B - SQRT(QUAD))/(2.*A)
               Y1 = T1*(YJ - YI) + YI
               Y2 = T2*(YJ - YI) + YI
         ENDIF
      IF ( Y1 .GT. FIREY .AND. Y1 .LT. R ) THEN
            WVFNS(I,K,II,KK) = 0.0
      ELSE IF( Y2 .GT. FIREY .AND. Y2 .LT. R) THEN
            WVFNS(I,K,II,KK) = 0.0
      ENDIF
C  END OF MODIFICATION TO WALL VIEW FACTORS WHEN THE FIRE IS INCLUDED
C************************************************************************

 350    WVFSN(II,KK,I,K) = WVFNS(I,K,II,KK) * DAREA1/ DAREA2
  300   CONTINUE
         WRITE(6,*)
C*************************************************************************
C     WVFNC  WALL VIEW FACTOR FROM NORTH SPHERE TO CYLINDER

      DO 400 I = NIS,NI-1
```

```
      DO 400 K = NKS,NA-1
      DO 400 II = NIS, NI-1
      DO 400 KK = NA,NB-1
      IF (I .EQ. II .AND. K .EQ. KK) THEN
         WVFNC(I,K,II,KK) = 0.0
      ELSE
         PHI1 = PHI(K) + .5*DPHIN
         RHO1 = R * SIN (PHI1)
         Z1 = Z(I,K)
         Z2 = Z(II,KK)
         THETAD = ABS( THETA(I,K) - THETA(II,KK))
         IF (THETAD.GT. PI) THETAD = 2*PI - THETAD
         DAREA1 = AREA(K-NKS+1)
         DAREA2 = AREAC
         ASQ = RHO1**2 + R**2 - 2.0*RHO1*R*COS(THETAD)
         BSQ = (Z1 -Z2)**2
         RSQ = ASQ + BSQ
         RD  = SQRT(RSQ)
         ZETA1S = BSQ + RHO1**2
         ZETA2S = R**2 + (Z2 -ZCYL1)**2
         CBETA1 = (R**2 + RSQ - ZETA1S)/ (2.0*R*RD)
         CBETA2 = (R**2 + RSQ - ZETA2S)/ (2.0*R*RD)
         WVFNC(I,K,II,KK) = (CBETA1*CBETA2)/(PI*RSQ) * DAREA2
      ENDIF
C***********************************************************************
C THE FOLLOWING SECTION IS ONLY INCLUDED IF THE FIRE IS CONSIDERED.
C IN THAT CASE,IFIRE (INCLUDED FIRE) WOULD EQUAL 1 AND THE CHECK TO
C SEE IF THE VIEW FACTOR INTERSECTS THE FIRE WOULD BE ACCOMPLISHED.
C************************************************************************
      IF (IFIRE .EQ. 0) GO TO 450
      IF(WVFNC(I,K,II,KK) .EQ. 0.) GO TO 450
      FIREY = -R + (R/MJ * FPAND)
      RF = DIAFP / 2.0
      XI = RHO1*COS(THETA(I,K))
      YI = RHO1*SIN(THETA(I,K))
      ZI = Z(I,K)
      XJ = R*COS(THETA(II,KK))
      YJ = R*SIN(THETA(II,KK))
      ZJ = Z(II,KK)
      XD = XJ - XI
      YD = YJ - YI
      ZD = ZJ - ZI
      A  = XD**2 + ZD**2
      B  = 2.0 *(ZI - HSZ)*ZD + 2.0*XI*XD
      C  = XI**2 + (ZI - HSZ)**2 - RF**2
      QUAD = B**2 - 4.*A*C
          IF ( QUAD .LT. 0.) THEN
               GO TO 450
          ELSE
               T1 = (-B + SQRT(QUAD))/(2.*A)
               T2 = (-B - SQRT(QUAD))/(2.*A)
               Y1 = T1*(YJ - YI) + YI
               Y2 = T2*(YJ - YI) + YI
          ENDIF
      IF ( Y1 .GT. FIREY .AND. Y1 .LT. R ) THEN
           WVFNC(I,K,II,KK) = 0.0
      ELSE IF( Y2 .GT. FIREY .AND. Y2 .LT. R) THEN
           WVFNC(I,K,II,KK) = 0.0
      ENDIF
C  END OF MODIFICATION TO WALL VIEW FACTORS WHEN THE FIRE IS INCLUDED
C**********************************************************************
C        WRITE(6,*)I,K,II,KK,WVFNC(I,K,II,KK)
 450     WVFCN(II,KK,I,K) = WVFNC(I,K,II,KK) * DAREA1/ DAREA2
  400    CONTINUE
          WRITE(6,*)
C************************************************************************
C     WVFSC  WALL VIEW FACTOR FROM SOUTH SPHERE TO CYLINDER
```

```
      DO 500 I = NIS,NI-1
      DO 500 K = NB,NK-1
      DO 500 II = NIS, NI-1
      DO 500 KK = NA,NB-1
      IF (I .EQ. II .AND. K .EQ. KK) THEN
         WVFSC(I,K,II,KK) = 0.0
      ELSE
         PHI1 = PI - (PHI(K) + .5*DPHIN)
         RHO1 = R * SIN (PHI1)
         H1 = R * COS(PHI1)
         Z1 = Z(I,K)
         Z2 = Z(II,KK)
         THETAD = ABS( THETA(I,K) - THETA(II,KK))
         IF (THETAD.GT. PI) THETAD = 2*PI - THETAD
         DAREA1 = AREA(K+MKN+1-NB)
         DAREA2 = AREAC
         ASQ = RHO1**2 + R**2 - 2.0*RHO1*R*COS(THETAD)
         BSQ = (Z1 -Z2)**2
         RSQ = ASQ + BSQ
         RD  = SQRT(RSQ)
         ZETA1S = BSQ + RHO1**2
         ZETA2S = R**2 + (Z2 -ZCYL2)**2
         CBETA1 = (R**2 + RSQ - ZETA1S)/ (2.0*R*RD)
         CBETA2 = (R**2 + RSQ - ZETA2S)/ (2.0*R*RD)
         WVFSC(I,K,II,KK) = (CBETA1*CBETA2)/(PI*RSQ) * DAREA2
      ENDIF
C**********************************************************************
C THE FOLLOWING SECTION IS ONLY INCLUDED IF THE FIRE IS CONSIDERED.
C IN THAT CASE,IFIRE (INCLUDED FIRE) WOULD EQUAL 1 AND THE CHECK TO
C SEE IF THE VIEW FACTOR INTERSECTS THE FIRE WOULD BE ACCOMPLISHED.
C**********************************************************************
      IF (IFIRE .EQ. 0) GO TO 550
      IF(WVFSC(I,K,II,KK) .EQ. 0.) GO TO 550
      FIREY = -R + (R/MJ * FPAND)
      RF = DIAFP / 2.0
      XI = RHO1*COS(THETA(I,K))
      YI = RHO1*SIN(THETA(I,K))
      ZI = Z(I,K)
      XJ = R*COS(THETA(II,KK))
      YJ = R*SIN(THETA(II,KK))
      ZJ = Z(II,KK)
      XD = XJ - XI
      YD = YJ - YI
      ZD = ZJ - ZI
      A  = XD**2 + ZD**2
      B  = 2.0 *(ZI - HSZ)*ZD + 2.0*XI*XD
      C  = XI**2 + (ZI - HSZ)**2 - RF**2
      QUAD = B**2 - 4.*A*C
          IF ( QUAD .LT. 0.) THEN
               GO TO 550
          ELSE
               T1 = (-B + SQRT(QUAD))/(2.*A)
               T2 = (-B - SQRT(QUAD))/(2.*A)
               Y1 = T1*(YJ - YI) + YI
               Y2 = T2*(YJ - YI) + YI
          ENDIF
      IF ( Y1 .GT. FIREY .AND. Y1 .LT. R ) THEN
           WVFSC(I,K,II,KK) = 0.0
      ELSE IF( Y2 .GT. FIREY .AND. Y2 .LT. R) THEN
           WVFSC(I,K,II,KK) = 0.0
      ENDIF
C  END OF MODIFICATION TO WALL VIEW FACTORS WHEN THE FIRE IS INCLUDED
C**********************************************************************
 550    WVFCS(II,KK,I,K) = WVFSC(I,K,II,KK) * DAREA1/ DAREA2
  500   CONTINUE
         WRITE(6,*)

C**********************************************************************
```

```
C     WVFCC  WALL VIEW FACTOR FROM CYLINDER TO CYLINDER
       DO 600 I = NIS,NI-1
       DO 600 K = NA,NB-1
       DO 600 II = NIS, NI-1
       DO 600 KK = NA,NB-1
       IF (I .EQ. II .AND. K .EQ. KK) THEN
          WVFCC(I,K,II,KK) = 0.0
       ELSE
          Z1 = Z(I,K)
          Z2 = Z(II,KK)
          THETAD = ABS( THETA(I,K) - THETA(II,KK))
          IF (THETAD.GT. PI) THETAD = 2*PI - THETAD
          ASQ = 2.*R**2*(1.0-COS(THETAD))
          BSQ = (Z1 -Z2)**2
          RSQ = ASQ + BSQ
          RD  = SQRT(RSQ)
          ZETA1S = BSQ + R**2
          CBETA1 = (R**2 + RSQ - ZETA1S)/ (2.0*R*RD)
          CBETA2 = CBETA1
          WVFCC(I,K,II,KK) = (CBETA1*CBETA2)/(PI*RSQ) * AREAC
       ENDIF
C***********************************************************************
C THE FOLLOWING SECTION IS ONLY INCLUDED IF THE FIRE IS CONSIDERED.
C IN THAT CASE,IFIRE (INCLUDED FIRE) WOULD EQUAL 1 AND THE CHECK TO
C SEE IF THE VIEW FACTOR INTERSECTS THE FIRE WOULD BE ACCOMPLISHED.
C***********************************************************************
       IF (IFIRE .EQ. 0) GO TO 650
       IF(WVFCC(I,K,II,KK) .EQ. 0.) GO TO 650
       FIREY = -R + (R/MJ * FPAND)
       RF = DIAFP / 2.0
       XI = R*COS(THETA(I,K))
       YI = R*SIN(THETA(I,K))
       ZI = Z(I,K)
       XJ = R*COS(THETA(II,KK))
       YJ = R*SIN(THETA(II,KK))
       ZJ = Z(II,KK)
       XD = XJ - XI
       YD = YJ - YI
       ZD = ZJ - ZI
       A  = XD**2 + ZD**2
       B  = 2.0 *(ZI - HSZ)*ZD + 2.0*XI*XD
       C  = XI**2 + (ZI - HSZ)**2 - RF**2
       QUAD = B**2 - 4.*A*C
            IF ( QUAD .LT. 0.) THEN
                  GO TO 650
            ELSE
                  T1 = (-B + SQRT(QUAD))/(2.*A)
                  T2 = (-B - SQRT(QUAD))/(2.*A)
                  Y1 = T1*(YJ - YI) + YI
                  Y2 = T2*(YJ - YI) + YI
            ENDIF
       IF ( Y1 .GT. FIREY .AND. Y1 .LT. R ) THEN
             WVFCC(I,K,II,KK) = 0.0
       ELSE IF( Y2 .GT. FIREY .AND. Y2 .LT. R) THEN
             WVFCC(I,K,II,KK) = 0.0
       ENDIF
C  END OF MODIFICATION TO WALL VIEW FACTORS WHEN THE FIRE IS INCLUDED
C**********************************************************************
 650  CONTINUE
  600    CONTINUE
          WRITE(6,*)
       RETURN
       END

       SUBROUTINE VIEW
       COMMON/BL1/ NIS,NI,NKS,NK,NA,NB,MI,MK,MKN,MKS,MKC,CL,DTHETA,
     & DPHIN,DPHIS,DZ1,DZ2,DZ3,Z1,R,PI,ZCYL1,ZCYL2
```

```
      COMMON/BL2/ PHI(33),THETA(2:21,33),Z(2:21,33),AREA(10),AREAC
      COMMON/BL3/ MREGN1,MREGN2,MREGN3,IREGN1,IREGN2,IREGN3,KSM1,KSM2,
     & KSM3,KSM4,KSM5,KSM6
      COMMON/BL4/ VFMXR(579,579),DELY(2,12),RF
      COMMON/BL5/WVFNN(2:21,3:7,2:21,3:7),WVFSS(2:21,26:30,2:21,26:30),
     & WVFSN(2:21,26:30,2:21,3:7),WVFNC(2:21,3:7,2:21,8:25),
     & WVFCS(2:21,8:25,2:21,26:30), WVFSC(2:21,26:30,2:21,8:25),
     & WVFNS(2:21,3:7,2:21,26:30),WVFCN(2:21,8:25,2:21,3:7),
     &WVFCC(2:21,8:25,2:21,8:25)
      COMMON/BL7/ NJS,NJ,MJ,HSZ,FPAND,HSANG(2,12),Y(2,12),HSY,DIAFP,
     &VFHNS(2,12,2:21,3:7),VFHSS(2,12,2:21,26:30),VFHC(2,12,2:21,8:25),
     &VFNSH(2:21,3:7,2,12),VFSSH(2:21,26:30,2,12),VFCH(2:21,8:25,2,12)
      COMMON/BLK8/VFMXC(579,579),VFMXIN(579,579),
     & CONSRA, NHSZ,AR(579),EM(579),IFIRE
*************************************************************************
*     VFMXR    =   VIEW FACTOR MATRIX , VIEW FACTORS BEFORE MODIFICATION*
*                  THIS IS A 579X579 MATRIX FOR THIS RUN                *
*     KSM1-6   =   DUMMY VARIABLES USED TO NUMBER THE CELLS FROM 1-579  *
*************************************************************************
C     SET UP VIEW FACTOR COEFFICIENT MATRIX, VFMXR
C     I,K IS THE CELL NUMBER STARTING FROM
C     II,KK IS THE CELL NUMBER GOING TO
C     VFMXR FOR NORTH SPHERE TO NORTH SPHERE, CYLINDER, SOUTH SPHERE
      KSM1 = MREGN1
      DO 5 K = NKS, NA-1
      DO 5 I = NIS, NI-1
          KSM1 = KSM1 + 1
          KSM2 = 1
          KSM3 = MREGN2
          KSM5 = MREGN3
C     VFMXR FOR NORTH SPHERE TO NORTH SPHERE
      DO 10 KK = NKS, NA-1
      DO 10 II = NIS, NI-1
          VFMXR(KSM1,KSM2) = WVFNN(I,K,II,KK)
C       WRITE(6,*) KSM1,KSM2, VFMXR(KSM1,KSM2)
          KSM2 = KSM2 + 1
   10  CONTINUE
C     VFMXR FOR NORTH SPHERE TO CYLINDER
      DO 15 KK = NA, NB-1
      DO 15 II = NIS, NI-1
          VFMXR(KSM1,KSM3) = WVFNC(I,K,II,KK)
          VFMXR(KSM3,KSM1) = WVFCN(II,KK,I,K)
C       WRITE(6,*) KSM1,KSM3, VFMXR(KSM1,KSM3),VFMXR(KSM3,KSM1)
          KSM3 = KSM3 + 1
   15  CONTINUE
C     VFMXR FOR NORTH SPHERE TO SOUTH SPHERE
      DO 20 KK = NB,NK-1
      DO 20 II = NIS, NI-1
          VFMXR(KSM1,KSM5) = WVFNS(I,K,II,KK)
          VFMXR(KSM5,KSM1) = WVFSN(II,KK,I,K)
C       WRITE(6,*) KSM1,KSM5, VFMXR(KSM1,KSM5),VFMXR(KSM5,KSM1)
          KSM5 = KSM5 + 1
   20  CONTINUE
   5  CONTINUE
C     VFMXR FOR CYLINDER TO __, CYLINDER, SOUTH SPHERE
      KSM3 = IREGN1
      DO 25 K = NA, NB-1
      DO 25 I = NIS, NI-1
          KSM3 = KSM3 + 1
          KSM4 = MREGN2
          KSM5 = MREGN3
C     VFMXR FOR CYLINDER TO CYLINDER
```

```
      DO 30 KK = NA, NB-1
      DO 30 II = NIS, NI-1
         VFMXR(KSM3,KSM4) = WVFCC(I,K,II,KK)
C        WRITE(6,*) KSM3,KSM4, VFMXR(KSM3,KSM4)
         KSM4 = KSM4 + 1
  30  CONTINUE
C     VFMXR FOR CYLINDER TO SOUTH SPHERE
      DO 35  KK = NB, NK-1
      DO 35  II = NIS, NI-1
         VFMXR(KSM3, KSM5) = WVFCS(I,K,II,KK)
         VFMXR(KSM5, KSM3) = WVFSC(II,KK,I,K)
C        WRITE(6,*) KSM3,KSM5, VFMXR(KSM3,KSM5),VFMXR(KSM5,KSM3)
         KSM5 = KSM5 + 1
  35  CONTINUE
  25  CONTINUE
C     VFMXR FOR SOUTH SPHERE TO ___, ____, SOUTH SPHERE
      KSM5 = IREGN2
      DO 40 K = NB, NK-1
      DO 40 I = NIS, NI-1
         KSM5 = KSM5 + 1
         KSM6 = MREGN3
C     VFMXR FOR SOUTH SPHERE TO SOUTH SPHERE
      DO 45 KK = NB, NK-1
      DO 45 II = NIS, NI-1
         VFMXR(KSM5, KSM6) = WVFSS( I,K,II,KK)
C        WRITE(6,*) KSM5,KSM6, VFMXR(KSM5,KSM6)
         KSM6 = KSM6 + 1
  45  CONTINUE
  40  CONTINUE
      RETURN
      END

      SUBROUTINE HEAT
      COMMON/BL1/ NIS,NI,NKS,NK,NA,NB,MI,MK,MKN,MKS,MKC,CL,DTHETA,
     & DPHIN,DPHIS,DZ1,DZ2,DZ3,Z1,R,PI,ZCYL1,ZCYL2
      COMMON/BL2/ PHI(33),THETA(2:21,33),Z(2:21,33),AREA(10),AREAC
      COMMON/BL3/ MREGN1,MREGN2,MREGN3,IREGN1,IREGN2,IREGN3,KSM1,KSM2,
     & KSM3,KSM4,KSM5,KSM6
      COMMON/BL4/ VFMXR(579,579),DELY(2,12),RF
      COMMON/BL5/WVFNN(2:21,3:7,2:21,3:7),WVFSS(2:21,26:30,2:21,26:30),
     & WVFSN(2:21,26:30,2:21,3:7),WVFNC(2:21,3:7,2:21,8:25),
     & WVFCS(2:21,8:25,2:21,26:30), WVFSC(2:21,26:30,2:21,8:25),
     & WVFNS(2:21,3:7,2:21,26:30),WVFCN(2:21,8:25,2:21,3:7),
     &WVFCC(2:21,8:25,2:21,8:25)
      COMMON/BL7/ NJS,NJ,MJ,HSZ,FPAND,HSANG(2,12),Y(2,12),HSY,DIAFP,
     &VFHNS(2,12,2:21,3:7),VFHSS(2,12,2:21,26:30),VFHC(2,12,2:21,8:25),
     &VFNSH(2:21,3:7,2,12),VFSSH(2:21,26:30,2,12),VFCH(2:21,8:25,2,12)
      COMMON/BLK8/VFMXC(579,579),VFMXIN(579,579),
     & CONSRA, NHSZ,AR(579),EM(579),IFIRE
      DIMENSION ASUM(2,12),COSUMN(2,12),COSUMS(2,12),FN(2,12),FS(2,12),
     & CBN(2,12,2:21,3:7),CBS(2,12,2:21,26:30),CBC(2,12,2:21,8:25),
     & ARN(2,12,2:21,3:7),ARS(2,12,2:21,26:30),ARC(2,12,2:21,8:25),
     & SVFN(2,12),SVFS(2,12),RDM(2,12,2:21,3:30),DIST(560,561:579)
*****************************************************************************
*   HSANG(I,J) =    HEAT SOURCE ANGLE, IT IS EITHER 90 OR 270 DEGREES    *
*   Y(I,J)     =    FIRE CELL Y LOCATION,  ALL CELLS ABOVE THE X AXIS    *
*                   ARE POSITVE AND ALL CELLS BELOW ARE NEGATIVE         *
*   RDM(I,J,II,KK) = ARRAY USED TO STORE THE DISTANCES BETWEEN THE FIRE *
*                     AND THE TANK CELLS                                 *
*   AREAR      =     THE RECTANGULAR AREA OF THE FIRE                    *
*   AREACI     =    CIRCULAR AREA OF THE FIRE (PI * RF**2)               *
*   CBN        =    FOUR INDICE ARRAY USED TO STORE THE CBETA1 VALUES    *
*                   FOR THE NORTH SPHERE, WHERE THE ORIGINATING CELL IS *
```

```
*                       THE FIRE CELL                                     *
*    CBS        =       FOUR INDICE ARRAY FOR THE SOUTH SPHERE            *
*    CBC        =       FOUR INDICE ARRAY FOR THE CYLINDER                *
*    ARN        =       FOUR INDICE ARRAY TO STORE THE AREA RATIO,        *
*                       AREA OF THE FIRE / AREA OF THE TANK ELEMENT. THIS IS*
*                       FOR THE NORTH SPHERE.  THE AREA OF THE FIRE IS A  *
*                       COMBINATION OF AREAR AND AREACI                   *
*    ARS        =       FOUR INDICE ARRAY TO STORE THE AREA RATIO FOR THE *
*                       SOUTH SPHERE                                      *
*    ARC        =       FOUR INDICE ARRAY TO STORE THE AREA RATIO FOR THE *
*                       CYLINDER                                          *
*    SVFN(I,J) =        ARRAY TO STORE THE SUM OF ALL THE VIEW FACTORS    *
*                       FROM A FIRE CELL TO ALL THE CELLS ON THE NORTH SIDE *
*                       OF THE TANK, CELLS 1-280                          *
*    SVFS(I,J) =        ARRAY TO STORE THE SUM OF ALL THE VIEW FACTORS FROM *
*                       A FIRE CELL TO ALL THE CELLS ON THE SOUTH SIDE OF *
*                       THE TANK, CELLS 281 - 560                         *
*    COSUMN(I,J) =      THE ARRAY TO STORE THE SUM: (1. - CBETA1)*VF(HEAT *
*                       SOURCE TO A TANK CELL ON THE NORTH SIDE) FOR A    *
*                       PARTICULAR FIRE CELL                              *
*    COSUMS(I,J) =      THE ARRAY TO STORE THE SIMILIAR VALUE FOR THE FIRE *
*                       CELL TO THE SOUTH TANK CELLS                      *
*    FN(I,J)    =       CORRECTION FACTOR FOR A FIRE CELL TO THE CELLS ON *
*                       THE NORTH SIDE                                    *
*    FS(I,J)    =       CORRECTION FACTOR FOR A FIRE CELL TO THE CELLS ON *
*                       THE SOUTH SIDE                                    *
*                                                                         *
*    VFHNS      =       VIEW FACTOR FROM THE HEAT SOURCE TO THE NORTH SPHERE*
*    VFNSH      =       VIEW FACTOR FROM THE NORTH SPHERE TO THE HEAT SOURCE*
*    VFHSS      =       VIEW FACTOR FROM THE HEAT SOURCE TO THE SOUTH SPHERE*
*    VFSSH      =       VIEW FACTOR FROM THE SOUTH SPHERE TO THE HEAT SOURCE*
*    VFHC       =       VIEW FACTOR FROM THE HEAT SOURCE TO THE CYLINDER  *
*    VFCH       =       VIEW FACTOR FROM THE CYLINDER TO THE HEAT SOURCE  *
*                                                                         *
*    FINDING THE VIEW FACTORS IS SIMILAR TO THE "WALL" SUBROUTINE AND     *
*    THE VARIABLES USED HERE HAVE THE SAME MEANING AS THOSE FOUND IN      *
*    IN THAT SUBROUTINE.                                                  *
***************************************************************************

C FIND THE LOCATION OF THE FIRE CELLS, ANGLE AND Y LOCATION
       DO 200 I = 1,2
       DO 200 J = NJS, NJ-1
       DELY(I,J) = R/MJ
 200  CONTINUE
       DO 210 I = 1,2
           Y1 = 0.0
       DO 210 J = NJS,NJ-1
           IF ( I .EQ. 1 ) THEN
                 HSANG(I,J) = PI/2.0
                 Y(I,J) = Y1 + DELY(I,J)/2.0
                 Y1 = Y(I,J) + DELY(I,J)/2.0
           ELSE
                 HSANG(I,J) = 3.0 * PI / 2.0
                 Y(I,J) = Y1 - DELY(I,J)/2.0
                 Y1 = Y(I,J) - DELY(I,J)/2.0
           ENDIF
  210 CONTINUE
C     WRITE(*,*)'I  ','J   ','Y   ','THETA   '
C     DO 215 I = 1,2
C     DO 215 J = NJS,NJ-1
C        WRITE(*,*) I,J,Y(I,J), HSANG(I,J)
C 215 CONTINUE
*************************************************************************


C************************************************************************
C  HEAT SOURCE VIEW FACTOR FROM THE HEAT SOURCE TO THE NORTH SPHERE
       M = NJ-1
       DO 220 I = 1,2
```

```
          IF(I .EQ. 2) M = MJ - FPAND
      DO 220 J = NJS, M
          SVFN(I,J) =0.0
          COSUMN(I,J) = 0.0
      DO 220 II = NIS, NI-1
      DO 220 KK = NKS, NA-1
          PHI1 = PHI(KK) + 0.5*DPHIN
          RHO1 = R * SIN (PHI1)
          H1   = R * COS (PHI1)
          DAREA2 = AREA(KK-NKS + 1)
          HSY = Y(I,J)
          EANG = ABS(HSANG(I,J) - THETA(II,KK) )
          IF(EANG .GT. PI)EANG = 2.0*PI - EANG
          ASQ = RHO1**2 + HSY**2 - 2.0*RHO1*(ABS(HSY))*COS(EANG)
          ZDIFF = HSZ - ZCYL1
          BSQ = (ZDIFF + H1)**2
          RSQ = ASQ + BSQ
          RD = SQRT(RSQ)
          RDM(I,J,II,KK) = RD
          ZETA2S = ZDIFF**2 + HSY**2
          CBETA2 = (R**2 + RSQ - ZETA2S) / (2.0 * R * RD)
          B = SQRT(BSQ)
          CBETA1 = B/RD
          CBN(I,J,II,KK)=CBETA1
          VFHNS(I,J,II,KK) = ((CBETA1*CBETA2)/(PI * RSQ)) * DAREA2
C FIND THE AREA THAT THE TANK ELEMENT "SEES OF THE FIRE"
          AREAR = DIAFP * DELY(I,J)
          AREACI = PI *(DIAFP/2.0)**2
          DAREA1 = AREAR*(1.-CBETA1) + AREACI*(CBETA1)
C USE RECIPROSITY
          VFNSH(II,KK,I,J) = VFHNS(I,J,II,KK) * DAREA1/DAREA2
C USED TO FIND MODIFICATION FACTOR
          ARN(I,J,II,KK) = DAREA1/DAREA2
          SVFN(I,J) = VFHNS(I,J,II,KK) + SVFN(I,J)
          COSUMN(I,J) = (1.-CBETA1)*VFHNS(I,J,II,KK)+
     &    COSUMN(I,J)
 220  CONTINUE
C***********************************************************************
C  HEAT SOURCE VIEW FACTOR FROM THE HEAT SOURCE TO THE SOUTH SPHERE
      M = NJ - 1
      DO 230 I = 1,2
          IF(I .EQ. 2) M = MJ - FPAND
      DO 230 J = NJS, M
          SVFS(I,J) =0.0
          COSUMS(I,J) = 0.0
      DO 230 II = NIS, NI-1
      DO 230 KK = NB, NK-1
          PHI1 = PI - (PHI(KK) + 0.5*DPHIS)
          RHO1 = R * SIN (PHI1)
          H1   = R * COS (PHI1)
          DAREA2 = AREA(KK+MKN+1-NB)
          HSY = Y(I,J)
          EANG = ABS(HSANG(I,J) - THETA(II,KK) )
          IF(EANG .GT. PI)EANG = 2.0*PI - EANG
          ASQ = RHO1**2 + HSY**2 - 2.0*RHO1*(ABS(HSY))*COS(EANG)
          ZDIFF = ZCYL2 - HSZ
          BSQ = (ZDIFF + H1)**2
          RSQ = ASQ + BSQ
          RD = SQRT(RSQ)
          RDM(I,J,II,KK) = RD
          ZETA2S = ZDIFF**2 + HSY**2
          CBETA2 = (R**2 + RSQ - ZETA2S) / (2.0 * R * RD)
          B = SQRT(BSQ)
          CBETA1 = B/RD
          CBS(I,J,II,KK)=CBETA1
          VFHSS(I,J,II,KK) = ((CBETA1*CBETA2)/(PI * RSQ))*DAREA2
C FIND THE AREA THAT THE TANK ELEMENT "SEES OF THE FIRE"
          AREAR = DIAFP * DELY(I,J)
```

```
           AREACI = PI *(DIAFP/2.0)**2
           DAREA1 = AREAR*(1.-CBETA1) + AREACI*(CBETA1)
C USE RECIPROSITY
           VFSSH(II,KK,I,J) = VFHSS(I,J,II,KK) * DAREA1/DAREA2
C USED TO FIND THE MODIFICATION FACTOR
           ARS(I,J,II,KK) = DAREA1/DAREA2
           SVFS(I,J) = VFHSS(I,J,II,KK) + SVFS(I,J)
           COSUMS(I,J) = (1.-CBETA1)*VFHSS(I,J,II,KK)+
     &     COSUMS(I,J)
 230  CONTINUE

C**********************************************************************
C  HEAT SOURCE VIEW FACTOR FROM THE HEAT SOURCE TO THE CYLINDER
       M = NJ - 1
       DO 240 I = 1,2
           IF(I .EQ. 2) M = MJ - FPAND
       DO 240 J = NJS, M
       DO 240 II = NIS, NI-1
       DO 240 KK = NA, NB-1
           DAREA2 = AREAC
           Z1 = Z(II,KK)
           HSY = Y(I,J)
           EANG = ABS(HSANG(I,J) - THETA(II,KK))
           IF(EANG .GT. PI)EANG = 2.0*PI - EANG
           ASQ = R**2 + HSY**2 - 2.0*R*(ABS(HSY))*COS(EANG)
           ZDIFF = ABS(HSZ - Z1)
           BSQ = ZDIFF**2
           RSQ = ASQ + BSQ
           RD = SQRT(RSQ)
           RDM(I,J,II,KK) = RD
           ZETA2S = ZDIFF**2 + HSY**2
           CBETA2 = (R**2 + RSQ - ZETA2S) / (2.0 * R * RD)
           B = SQRT(BSQ)
           CBETA1 = B/RD
           CBC(I,J,II,KK)=CBETA1
           VFHC(I,J,II,KK) = ((CBETA1*CBETA2)/(PI * RSQ))*DAREA2
C FIND THE AREA THAT THE TANK ELEMENT "SEES OF THE FIRE"
           AREAR = DIAFP * DELY(I,J)
           AREACI = PI *(DIAFP/2.0)**2
           DAREA1 = AREAR*(1.-CBETA1) + AREACI*(CBETA1)
C USE RECIPROSITY
           VFCH(II,KK,I,J) = VFHC(I,J,II,KK) * DAREA1/DAREA2
C USED TO FIND THE MODIFICATION FACTOR
           ARC(I,J,II,KK) = DAREA1/DAREA2
C NORTH CELLS
           IF ( KK .LT.((NB-NA)/2. + NA))THEN
                 SVFN(I,J) = SVFN(I,J) + VFHC(I,J,II,KK)
                 COSUMN(I,J) =(1.-CBETA1)*VFHC(I,J,II,KK) + COSUMN(I,J)
C SOUTH CELLS
           ELSE IF ( KK .GE.((NB-NA)/2. + NA)) THEN
                 SVFS(I,J) = SVFS(I,J) + VFHC(I,J,II,KK)
                 COSUMS(I,J) =(1.-CBETA1)*VFHC(I,J,II,KK) + COSUMS(I,J)
           ENDIF
 240  CONTINUE
C**********************************************************************
C FIND THE MODIFICATION FACTOR FOR THE FIRE CELLS
C NOTE THAT THIS OUTER DO LOOP IS USED TO MODIFY THE FN/FS VALUES
C FURTHER ITERATIONS IMPROVE THE ACCURACY OF THIS MODIFICATION ROUTINE
C FROM PRIOR TESTING FOR THIS CASE, TWO ITERATIONS ARE SUFFICIENT
       DO 246 N = 1,2
       M = NJ -1
       DO 250 I = 1,2
           IF(I .EQ. 2) M = MJ - FPAND
       DO 250 J = NJS,M
            FN(I,J) = ( 1. -SVFN(I,J))/ COSUMN(I,J)
            FS(I,J) = ( 1. -SVFS(I,J)) / COSUMS(I,J)
 250  CONTINUE
```

```
C  USE THIS FN/FS VALUE TO MODIFY THE VIEW FACTOR FROM THE FIRE CELL
C  TO THE TANK CELL, THEN CALCULATE A NEW FN/FS BY USING SVFN/COSUMN
C  MATRICES THAT ARE NOW MODIFIED AGAIN.

       M = NJ -1
       DO 255 I = 1,2
          IF(I .EQ. 2) M = MJ - FPAND
       DO 255 J = NJS,M
       COSUMN(I,J) = 0.
       COSUMS(I,J) = 0.
       SVFN(I,J) = 0.
       SVFS(I,J) = 0.
       DO 255 II = NIS, NI-1
       DO 255 KK = NKS,NK-1
C THE NORTH SPHERE
        IF ( KK .LT. NA) THEN
        VFHNS(I,J,II,KK)=VFHNS(I,J,II,KK)*(1.+FN(I,J)*
     & (1.-CBN(I,J,II,KK)))
C   USE RECIPROSITY
        VFNSH(II,KK,I,J) = VFHNS(I,J,II,KK)*ARN(I,J,II,KK)
C   FIND A NEW SVF/COSUM
        SVFN(I,J) = VFHNS(I,J,II,KK) + SVFN(I,J)
        COSUMN(I,J)=(1.-CBN(I,J,II,KK))*VFHNS(I,J,II,KK) + COSUMN(I,J)
C  NORTH SIDE OF THE CYLINDER CELLS
        ELSE IF (KK .LT. ((NB-NA)/2  +NA)) THEN
        VFHC(I,J,II,KK)=VFHC(I,J,II,KK)*(1.+FN(I,J)*
     & (1.-CBC(I,J,II,KK)))
C   USE RECIPROSITY
        VFCH(II,KK,I,J) = VFHC(I,J,II,KK)*ARC(I,J,II,KK)
C   FIND A NEW SVF/COSUM
        SVFN(I,J) = VFHC(I,J,II,KK) + SVFN(I,J)
        COSUMN(I,J)=(1.-CBC(I,J,II,KK))*VFHC(I,J,II,KK) + COSUMN(I,J)
C  SOUTH SIDE OF THE CYLINDER CELLS
        ELSE IF (KK .GE. ((NB-NA)/2  +NA) .AND. KK .LT.NB ) THEN
        VFHC(I,J,II,KK)=VFHC(I,J,II,KK)*(1.+FS(I,J)*
     & (1.-CBC(I,J,II,KK)))
C   USE RECIPROSITY
        VFCH(II,KK,I,J) = VFHC(I,J,II,KK)*ARC(I,J,II,KK)
C   FIND A NEW SVF/COSUM
        SVFS(I,J) = VFHC(I,J,II,KK) + SVFS(I,J)
        COSUMS(I,J)=(1.-CBC(I,J,II,KK))*VFHC(I,J,II,KK) + COSUMS(I,J)
        ELSE
C   SOUTH SPHERE
        VFHSS(I,J,II,KK)=VFHSS(I,J,II,KK)*(1.+FS(I,J)*
     & (1.-CBS(I,J,II,KK)))
C   USE RECIPROSITY
        VFSSH(II,KK,I,J) = VFHSS(I,J,II,KK)*ARS(I,J,II,KK)
C   FIND A NEW SVF/COSUM
        SVFS(I,J) = VFHSS(I,J,II,KK) + SVFS(I,J)
        COSUMS(I,J)=(1.-CBS(I,J,II,KK))*VFHSS(I,J,II,KK) + COSUMS(I,J)
        ENDIF
 255  CONTINUE
 246  CONTINUE

************************************************************************

C CONVERT THE VIEW FACTORS INTO THE VFMXR MATRIX.  THE FIRE CELLS WILL
C BE NUMBERED FROM THE FIRE PAN TO THE TOP OF THE CYLINDER.  (561-579)
C CHANGE THE FOUR INDICE ARRAY FOR THE DISTANCE BETWEEN THE CELLS TO A
C TWO INDICE ARRAY, CALLED DIST(1,2). NOTE THE VIEW FACTORS FROM THE FIRE
C TO THE TANK ARE DIVIDED BY TWO BECAUSE OF THE TWO SIDEDNESS OF THE
C OF THE FIRE
       KSM1 = IREGN3
       DO 260 I = 1,2
          L = 2 + 1 -I
          M = NJ - 1
```

```
          IF (L .EQ. 2) M = MJ - FPAND
      DO 260 J = NJS, M
          IF ( L.EQ. 2) THEN
            JJ = M+NJS -J
          ELSE
            JJ = J
          ENDIF
          KSM1 = KSM1 + 1
          KSM2 = 1
          KSM3 = MREGN2
          KSM4 = MREGN3

C FROM THE HEAT SOURCE TO THE NORTH SPHERE
      DO 265 KK = NKS, NA-1
      DO 265 II = NIS,NI-1
          VFMXR(KSM1,KSM2) = VFHNS(L,JJ,II,KK)/2.0
C USE RECIPROSITY
          VFMXR(KSM2,KSM1) = VFNSH(II,KK,L,JJ)
          DIST(KSM2,KSM1) = RDM(L,JJ,II,KK)
          KSM2 = KSM2 + 1
  265 CONTINUE
C FROM THE HEAT SOURCE TO THE CYLINDER
      DO 270 KK = NA, NB-1
      DO 270 II = NIS, NI-1
          VFMXR(KSM1,KSM3) = VFHC(L,JJ,II,KK)/2.0
C USE RECIPROSITY
          VFMXR(KSM3,KSM1) = VFCH(II,KK,L,JJ)
          DIST(KSM3,KSM1) = RDM(L,JJ,II,KK)
          KSM3 = KSM3 + 1
  270 CONTINUE
C FROM THE HEAT SOURCE TO THE SOUTH SPHERE
      DO 275 KK = NB, NK-1
      DO 275 II = NIS, NI-1
          VFMXR(KSM1,KSM4) = VFHSS(L,JJ,II,KK)/2.0
C USE RECIPROSITY
          VFMXR(KSM4,KSM1) = VFSSH(II,KK,L,JJ)
          DIST(KSM4,KSM1) = RDM(L,JJ,II,KK)
          KSM4 =KSM4 + 1
  275 CONTINUE
  260 CONTINUE
      DO 276 I = 561,579
      DO 276 J = 561,579
          VFMXR(I,J) = 0.0
  276 CONTINUE
C***********************************************************************
C THE FOLLOWING SECTION CORRECTS FOR THE VIEW FACTORS FROM THE TANK
C TO THE FIRE.  SINCE THE EXACT AREA OF THE FIRE IS NOT KNOWN THERE
C MAY BE AN ERROR IN THE CALCULATION.  THE TOTAL SUM OF THE VIEW FACTOR
C FROM ONE CELL TO EVERYTHING IN THE TANK MUST EQUAL ONE.  THEREFORE
C THE VIEW FACTORS FROM THE TANK TO THE FIRE MUST BE MODIFIED AS
C AS FOLLOWS:
C VARIABLES      VFTSUM = VIEW FACTOR TOTAL SUM, FROM ONE CELL TO ALL
C                         THE OTHER CELLS IN THE TANK INCLUDING THE FIRE
C                DENOM = DENOMINATOR OF THE MODIFCATION WHICH IS A
C                        SUM OVER THE FIRE CELLS
C                A     = MODIFICATION FACTOR
C

      DO 280 I = 1, 560
          VFTSUM = 0.
          DENOM = 0.
      DO 285 J = 1, 579
          VFTSUM = VFTSUM + VFMXR(I,J)
          IF( J .GE. 561) THEN
                DENOM = DENOM + VFMXR(I,J)/(SQRT(1.+(DIST(I,J)/RF)**2))
          ENDIF
  285 CONTINUE
```

```
         A = (1.-VFTSUM) / DENOM
         DO 290 K = 561,579
         VFMXR(I,K) = (1.+ A/(SQRT(1.+ (DIST(I,K)/RF)**2)))*VFMXR(I,K)
  290 CONTINUE
  280 CONTINUE
      RETURN
      END

      SUBROUTINE INVER
      COMMON/BL1/ NIS,NI,NKS,NK,NA,NB,MI,MK,MKN,MKS,MKC,CL,DTHETA,
     & DPHIN,DPHIS,DZ1,DZ2,DZ3,Z1,R,PI,ZCYL1,ZCYL2
      COMMON/BL2/ PHI(33),THETA(2:21,33),Z(2:21,33),AREA(10),AREAC
      COMMON/BL3/ MREGN1,MREGN2,MREGN3,IREGN1,IREGN2,IREGN3,KSM1,KSM2,
     & KSM3,KSM4,KSM5,KSM6
      COMMON/BL4/ VFMXR(579,579),DELY(2,12),RF
      COMMON/BL5/WVFNN(2:21,3:7,2:21,3:7),WVFSS(2:21,26:30,2:21,26:30),
     & WVFSN(2:21,26:30,2:21,3:7),WVFNC(2:21,3:7,2:21,8:25),
     & WVFCS(2:21,8:25,2:21,26:30), WVFSC(2:21,26:30,2:21,8:25),
     & WVFNS(2:21,3:7,2:21,26:30),WVFCN(2:21,8:25,2:21,3:7),
     &WVFCC(2:21,8:25,2:21,8:25)
      COMMON/BL7/ NJS,NJ,MJ,HSZ,FPAND,HSANG(2,12),Y(2,12),HSY,DIAFP,
     &VFHNS(2,12,2:21,3:7),VFHSS(2,12,2:21,26:30),VFHC(2,12,2:21,8:25),
     &VFNSH(2:21,3:7,2,12),VFSSH(2:21,26:30,2,12),VFCH(2:21,8:25,2,12)
      COMMON/BLK8/VFMXC(579,579),VFMXIN(579,579),
     & CONSRA, NHSZ,AR(579),EM(579),IFIRE
************************************************************************
*    WKAREA     =     WORK SPACE REQUIRED BY THE IMSL ROUTINE          *
*    VFMXC      =     MATRIX MODIFIED BEFORE INVERSION AND THEN LATER  *
*                     USED TO STORE THE COEFICIENTS FROM THE INVERTED  *
*                     MATRIX TIMES THE RIGHT HAND MATRIX.  THE "G" MATRIX*
*    VFMXR      =     ORIGINAL MATRIX WITH THE VIEW FACTORS THEN       *
*                     MULTIPLIED BY -SIGMA TO GIVE THE RIGHT-HAND MATRIX *
*    VFMXIN     =     INVERTED VFMXC MATRIX FROM THE IMSL ROUTINE      *
*    SIGMA      =     STEFAN-BOLTZMAN CONSTANT                         *
*    EM(J)      =     EMMISIVITY                                       *
************************************************************************
      DIMENSION WKAREA(579)
C SIGMA IS SET TO ONE IN THIS PROGRAM, THE ACTUAL VALUE OF SIGMA WILL
C BE USED IN THE TANK PROGRAM.
C     SIGMA = 1.714E-9
      SIGMA = 1.0
      CONSRA = 1.
      MZ = IREGN3
      NZ = MZ + 19
      DO 10 J = 1, NZ
         IF ( J .LE. MZ ) THEN
              EM(J) = .84
         ELSE
              EM(J) = .81
         ENDIF
 10   CONTINUE
CCTO SAVE SPACE, THE LEFT HAND MATRIX WILL BE CALLED VFMXC AND
CCLATER WILL BE WRITTEN OVER
      DO 15 I = 1, NZ
      DO 15 J = 1, NZ
      VFMXC(I,J) =(EM(J) -1.)*VFMXR(I,J) / EM(J)/AR(J)
 15   CONTINUE
      DO 20 I = 1,NZ
      VFMXC(I,I) = (1.-VFMXR(I,I)*(1.-EM(I))) / (AR(I) *EM(I))
 20   CONTINUE
CCTHE RIGHT HAND MATRIX WILL BE CALLED VFMXR AND WILL REPLACE
CCTHE ORIGINAL VIEW FACTOR MATRIX
      DO 25 I = 1,NZ
      DO 25 J = 1,NZ
         VFMXR(I,J) = - VFMXR(I,J)*SIGMA
```

```
 25   CONTINUE
        DO 26 I = 1,NZ
            VFMXR(I,I) = SIGMA+VFMXR(I,I)
 26   CONTINUE
        NN = NZ
        MM = NZ
        IA = NZ
        IAIN = NZ
        IDGT = 3
        CALL LINV1F(VFMXC,IA,NN,VFMXIN,IDGT,WKAREA,IER)
C MULTIPLY THE INVERTED MATRIX BY THE RIGHT HAND MATRIX TO GET THE
C REQUIRED "G" MATRIX
        DO 30 I = 1, NZ
        DO 30 J = 1, NZ
            VFMXC(I,J) = 0.
        DO 30 K = 1, NZ
            VFMXC(I,J) = VFMXC(I,J) + VFMXIN(I,K)*VFMXR(K,J)
 30   CONTINUE
C THE FOLLOWING PRINT STATEMENTS CHECK A FEW ROWS TO SEE WHAT THE
C ELEMENTS ARE.  THESE PRINT STATEMENTS CAN BE OMITTED
            WRITE(6,*) 'I    ','J      ', 'VFMXC    '
        DO 50 I = 1,  2
        DO 50 J = 1, NZ
            WRITE(6,*) I,J, VFMXC(I,J)
 50   CONTINUE
        DO 51 J = 1, NZ
            I = 561
            WRITE(6,*) I,J, VFMXC(I,J)
 51   CONTINUE
C THIS DO LOOP SUMS UP THE ROW OF THE "G" MATRIX TO SEE IF IT GOES TO
C ZERO
        DO 56 I = 1, 579
             AA = 0.
        DO 55 J = 1, 579
             AA = VFMXC(I,J) + AA
C       IF (J .EQ. 560)WRITE(*,*)I,'SUM 560=',AA
 55   CONTINUE
        WRITE(*,*) 'ROW', I, 'SUM =', AA
 56   CONTINUE
C THESE STATEMENTS WRITE THE "G" MATRIX COEFICIENTS TO A DISK FOR
C USE WITH THE TANK PROGRAM
        WRITE(9) VFMXC
        REWIND 9
        RETURN
        END
        SUBROUTINE AREA1
        COMMON/BL1/ NIS,NI,NKS,NK,NA,NB,MI,MK,MKN,MKS,MKC,CL,DTHETA,
     & DPHIN,DPHIS,DZ1,DZ2,DZ3,Z1,R,PI,ZCYL1,ZCYL2
        COMMON/BL2/ PHI(33),THETA(2:21,33),Z(2:21,33),AREA(10),AREAC
        COMMON/BL3/ MREGN1,MREGN2,MREGN3,IREGN1,IREGN2,IREGN3,KSM1,KSM2,
     & KSM3,KSM4,KSM5,KSM6
        COMMON/BL4/ VFMXR(579,579),DELY(2,12),RF
        COMMON/BL5/WVFNN(2:21,3:7,2:21,3:7),WVFSS(2:21,26:30,2:21,26:30),
     & WVFSN(2:21,26:30,2:21,3:7),WVFNC(2:21,3:7,2:21,8:25),
     & WVFCS(2:21,8:25,2:21,26:30), WVFSC(2:21,26:30,2:21,8:25),
     & WVFNS(2:21,3:7,2:21,26:30),WVFCN(2:21,8:25,2:21,3:7),
     &WVFCC(2:21,8:25,2:21,8:25)
        COMMON/BL7/ NJS,NJ,MJ,HSZ,FPAND,HSANG(2,12),Y(2,12),HSY,DIAFP,
     &VFHNS(2,12,2:21,3:7),VFHSS(2,12,2:21,26:30),VFHC(2,12,2:21,8:25),
     &VFNSH(2:21,3:7,2,12),VFSSH(2:21,26:30,2,12),VFCH(2:21,8:25,2,12)
        COMMON/BLK8/VFMXC(579,579),VFMXIN(579,579),
     & CONSRA, NHSZ,AR(579),EM(579),IFIRE
C THIS SUBROUTINE ASSIGNS AN AREA TO A CELL BY THE CELL'S NUMBER
C     HS = 19
        DO 55 I = 1,IREGN3
```

```
          AR(I) = 0.0
 55   CONTINUE
       DO 60 I = 1 ,MKN
       DO 60 J = 1, IREGN1
          IF (J .LE. MI*I .AND. AR(J) .EQ. 0. ) AR(J) = AREA(I)
 60   CONTINUE
       DO 70 J = MREGN2,IREGN2
          AR(J) = AREAC
 70   CONTINUE
       DO 80 J = MREGN3,IREGN3
       DO 80 I = MKN+1,MKN+MKS
          IF(J .LE. (MI*(I-MKN)+IREGN2).AND.AR(J).EQ.0.)AR(J)=AREA(I)
 80   CONTINUE
C IT SETS THE FIRE AREA TO BE A RECTANGLE
       DY = R / MJ
       DO 85 K = 561, 579
          AR(K) = DIAFP * DY
 85   CONTINUE
       RETURN
       END
```

APPENDIX B

FORTRAN LISTING OF SPHERICAL/CYLINDRICAL NUMERICAL MODEL

```
C      ****************************************************************
C      **                                                            **
C      **          THREE-DIMENSIONAL NUMERICAL SIMULATION            **
C      **        OF A FIRE SPREAD INSIDE A NAVY STORAGE TANK         **
C      **                                                            **
C      **                    DEVELOPED BY :                          **
C      **               H.Q. YANG AND K.T. YANG                      **
C      **                                                            **
C      **     DEPARTMENT OF AEROSPACE & MECHANICAL ENGINEERING       **
C      **               UNIVERSITY OF NOTRE DAME                     **
C      **               NOTRE DAME, INDIANA, 46556                   **
C      **                                                            **
C      **                      DEC. 1986                             **
C      **                                                            **
C      ****************************************************************
C
       COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
       COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
       COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
       COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
       COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
       COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
       COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &            ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
       COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &            NCHPK(10),TCHP(10),CPS(10),CONS(10)
       COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &       ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
       COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
       COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
       COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
       COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
       COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
       COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)
       COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
       DIMENSION VFMXC(579,579),T4WALL(579)
       DATA N,ITLEFT,SORMAX,XTIME,ITMAX/20,400000,0.40,0.0,4/


C *** U0      : REFERENCE VELOCITY (FT/SEC),1 FT/SEC
C *** RHO0    : REFERENCE AIR DENSITY (LBM/FT**3)
C *** H       : REFERENCE LENGTH (FT)
C *** TA      : REFERENCE TEMPERATURE (R)
C *** TINIT   : INITIAL TEMPERATURE (0)
C *** GC      : GRAVITATIONAL CONSTANT
C *** RAITR   : GAS CONSTANT; 53.34
C *** CONST1  : RA*U0**2/GC
C *** CONST3  : INVERSE OF TA
```

```
C *** CONST4 : REFERENCE LENGTH (CM)
C *** CONST6 : REFERENCE VELOCITY (CM/S)
C *** CONSRA : TA**3/(RA*CP*U0*H*H)
C *** NTRWR  : NTREAL/NWRITE*NWRITE
C *** NTRWA  : NTREAL/NWALT*NWALT
C *** HCONV  : HEAT TRANSFER COEFFICIENT ON THE AMBIENT (BTU/H.FT**2K)
C
C
C
C *** RAD,H: RADIUS OF THE CYLINDRICAL AND SPHERICAL SECTIONS
C     CYL  : LENGTH OF THE CYLINDRICAL SECTION OF THE TANK
C *** NI   : TOTAL NUMBER CELLS IN  THETA-DIRECTION
C     NJ   :                        R-DIRECTION
C     NK   :                        Z AND PHI-DIRECTIONS
C     NA   :          FIRST NUMBER  Z-DIRECTION, ALONG THE CYLINDER AXIS
C     NB   :          LAST  NUMBER  Z-DIRECTION, ALONG THE CYLINDER AXIS
C *** HSZ(1,1),HSZ(1,2)  FIRST AND LAST COORDIANTE OF HEAT SOURCE
C                             IN X-DIRECTION  (IN DIMENSIONLESS FORM)
C     HSZ(2,1),HSZ(2,2)  FIRST AND LAST COORDIANTE OF HEAT SOURCE
C                             IN Y-DIRECTION  (IN DIMENSIONLESS FORM)
C     HSZ(3,1),HSZ(3,2)  FIRST AND LAST COORDIANTE OF HEAT SOURCE
C                             IN Z-DIRECTION  (IN DIMENSIONLESS FORM)
C
C *** ICHPB() : STARTING NODAL NUMBER FOR SOLID IN THETA-DIRECTION
C     JCHPB() :                                            R-DIRECTION
C     KCHPB() :                                      Z OR PHI-DIRECTION
C *** NCHPI() : NUMBER OF NODALS  FOR SOLID IN THETA-DIRECTION
C     NCHPJ() :                                        R-DIRECTION
C     NCHPK() :                                  Z,PHI-DIRECTION
C ****************************************************************
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C        INPUT DATA                                                         &
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
      CALL INPUT
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C        GENERATE GRID SYSTEM                                               &
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
      CALL GRID
C ###############################################################
C ***   READ VIEW FACTOR INVERSE MATRIX                           #
C################################################################
  999 READ (11,END=998) VFMXC
      GOTO 999
  998 CONTINUE
      REWIND 11
      CLOSE (11)
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C        INITIALIZE THE WHOLE FIELD                                         &
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
      CALL INIT
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C         START CALCULATION                                                 &
C &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

      NT=0
      NTIM=0
  300 CONTINUE
      NT=NT+1
C ***    NTMAX0 HAS THE MEANING AS "NTREAL" WHEN IT IS READ FROM
C        DISK OR TAPE.
      IF(XTIME .GT. TMAX) GO TO 303
      NTREAL=NT+NTMAX0
      TIME=TIME+DTIME
      XTIME=TIME*H/U0
```

```
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C     CALCULATE THE TRANSIENT HEAT INPUT                              &
C     NOTE IF 1 IN PARENTHESIS, THE BURN RATE IS CALCULATED           &
C     BY THE PRESSURE CURVE.  IF EQUAL TO TWO, THE BURN RATE          &
C     CURVE IS EITHER GIVEN OR ESTIMATED                              &
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
         CALL CALQ(2)
C ***    START CALCULATION
       ITER=0
       JTERM=0
       JJTERM=0
C     DEFINE THE UPDATED TPD(I,J,K), CPD(I,J,K),RPD(I,J,K)
C     UPD(I,J,K) AND VPD(I,J,K) FOR THE USE OF CALVIS AND SU(I,J,K)
       DO 48 K=1,NKP1
       DO 48 J=1,NJP1
       DO 48 I=1,NIP1
       TPD(I,J,K)=T(I,J,K)
       CPD(I,J,K)=C(I,J,K)
       RPD(I,J,K)=R(I,J,K)
       UPD(I,J,K)=U(I,J,K)
       VPD(I,J,K)=V(I,J,K)
       WPD(I,J,K)=W(I,J,K)
   48 CONTINUE
   29 CONTINUE
       JTERM=JTERM+1
  301 CONTINUE

C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C    CALCULATE THE RADIATION HEAT FLUX AT EVERY NRAD TIME STEPS &
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
       NRAD = 2
       IF (MOD(NT,NRAD).NE.0) GOTO 4000
       CALL RADHT(T4WALL,VFMXC)
 4000 CONTINUE
C####################################################################
C     CALCULATE THE TEMPERATURE                                      #
C####################################################################
       CALL CALT
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C     CALCULATE THE SMOKE CONCENTRATION                               &
C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C     CALL CALC
       DO 2000 J=1,NJP1
       DO 2000 I=1,NIP1
       DO 2000 K=1,NKP1
       IF(T(I,J,K).LT.TCOOL) T(I,J,K)=TCOOL
 2000 CONTINUE
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     GLOBLE PRESSURE CORRECTION FOR ENCLOSED TANK AIR               %
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        CALL GLOBE
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     CALCULATE THE TURBULENT VISCOSITY AND CONDUCTIVITY             @
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
       CALL CALVIS
C********************************************************************
C     CALCULATE THE DENSITY                                          *
C********************************************************************
       DO 100 J=1,NJP1
       DO 100 I=1,NIP1
       DO 100 K=1,NKP1
       IF (NOD(I,J,K).EQ.1) GOTO 100
       AAAA=BUOY*UGRT*HEIGHT(I,J,K)
       R(I,J,K)=(UGRT*P(I,J,K)+(1./EXP(AAAA)))/T(I,J,K)
```

```
  100 CONTINUE
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     CORRECT CONDUCTIVITY OF THE SOLID                                 $
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
       IF (NCHIP.EQ.0) GOTO 410
       CALL SOLCON
  410 CONTINUE
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     START PRESSURE CORRECTION ITERATIVE LOOP, IT IS THE MAJOR         %
C        PART OF THE ERROR CONTROL ROUTINE                              %
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

       ITER=ITER+1

C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C     CALCULATE THE VELOSITY U,V,AND W                                  @
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

       CALL CALU
CC    CALL STRESS
C *** **************
       CALL CALV
CC    CALL STRESS
C *** **************
       CALL CALW
CC    CALL STRESS
C *** **************

C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     CALCULATE THE PRESSURE AND STRESS                                 &
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

       CALL CALP
       CALL STRESS
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C      IF SOURCE TERM IS LARGER THAN 10.0, STOP PROGRAM                 %
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       IF (RESORM(ITER).GT.10.0) GOTO 2020

       IF(RESORM(ITER) .LE. SORMAX) GO TO 49
       IF(ITER .EQ. 1) GO TO 302
       ITERM1=ITER-1
       IF(RESORM(ITER) .LE. RESORM(ITERM1)) GO TO 302
       GO TO 304
  302 IF(JTERM .GE. 2) GO TO 37
       SOURCE=RESORM(ITER)
       GO TO 39
   37 IF(RESORM(ITER) .LE. SOURCE) GO TO 38
       GO TO 304
   38 SOURCE=RESORM(ITER)
   39 CONTINUE
       WRITE(6,95) ITER,RESORM(ITER),SORSUM
   95 FORMAT(53X,'ITER=',I2,2X,'SOURCE=',F9.6,2X,'SORMUP=',F9.6)
       DO 23 K=1,NKP1
       DO 23 J=1,NJP1
       DO 23 I=1,NIP1
       TPD(I,J,K)=T(I,J,K)
       CPD(I,J,K)=C(I,J,K)
       RPD(I,J,K)=R(I,J,K)
       UPD(I,J,K)=U(I,J,K)
       VPD(I,J,K)=V(I,J,K)
       WPD(I,J,K)=W(I,J,K)
       PPD(I,J,K)=P(I,J,K)
   23 CONTINUE
       JJTERM=0
       IF(ITER .EQ. ITMAX) GO TO 49
       IF(JTERM .EQ. 2) GO TO 35
       IF(ITER .EQ. 4) GO TO 29
   35 CONTINUE
       IF(JTERM .EQ. 3) GO TO 58
```

```
      IF(ITER .EQ. 7) GO TO 29
   58 CONTINUE
      JJTERM=0
      GO TO 301
  304 CONTINUE
      JJTERM=JJTERM+1
      IF(JJTERM .EQ. 1) WRITE(6,95) ITER,RESORM(ITER),SORSUM
      IF(JTERM .EQ. 1) GO TO 41
      IF(JTERM .EQ. 2 .AND. JJTERM .EQ. 1 .AND. ITER .NE. 5) GO TO 41
      GO TO 82
   41 CONTINUE
      DO 40 K=1,NKP1
      DO 40 J=1,NJP1
      DO 40 I=1,NIP1
      R(I,J,K)=RPD(I,J,K)
      U(I,J,K)=UPD(I,J,K)
      V(I,J,K)=VPD(I,J,K)
      W(I,J,K)=WPD(I,J,K)
      P(I,J,K)=PPD(I,J,K)
   40 CONTINUE
      IF(ITER .EQ. ITMAX) GO TO 49
      GO TO 29
   82 CONTINUE
      DO 43 K=1,NKP1
      DO 43 J=1,NJP1
      DO 43 I=1,NIP1
      T(I,J,K)=TPD(I,J,K)
      C(I,J,K)=CPD(I,J,K)
      R(I,J,K)=RPD(I,J,K)
      U(I,J,K)=UPD(I,J,K)
      V(I,J,K)=VPD(I,J,K)
      W(I,J,K)=WPD(I,J,K)
      P(I,J,K)=PPD(I,J,K)
   43 CONTINUE
      IF(ITER .EQ. ITMAX) GO TO 49
      IF((JTERM .EQ. 3 .AND. ITER .NE. 8) .OR. JJTERM .EQ. 2) GO TO 49
      GO TO 301
   49 CONTINUE
      ITERT=ITERT+ITER
C#################################################################
C     GO TO THE PRESSURE TRACKING SUBROUTINE ,PRINT OUT          #
C     RESULTS IF AT THE RIGHT TIME INTERVAL                      #
C#################################################################
      CALL PTRACK
      IF (MOD(NTREAL,NWRP).EQ.0) CALL OUT(1)
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C      FIND TEMPERATURES AT THERMOCOUPLE POINTS AND PRINT OUT    %
C      IF AT THE RIGHT TIME INTERVAL                             %
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      CALL TCP
      IF (MOD(NTREAL,NWRP).EQ.0) CALL OUT(2)
 2422 CONTINUE
      IF (MOD(NTREAL,NWRITE).EQ.0) CALL OUT(3)
C     IF(NTREAL .EQ. NTREAL/NWRITE*NWRITE) CALL OUT(3)
  505 CONTINUE
      IF((XTIME+DTIME*H/U0) .GE. TMAX) GO TO 277
C *** ***********************************************************
C     CALL TLEFT(IT)
C 123 FORMAT(' ITLEFT = ',I10)
C     IT0=IT
C     IF(IT.LT.ITLEFT) CALL OUT(3)
C *** ***********************************************************

C ***    RESET THE OLD TIME VALUES  TOD, ROD, UOD, VOD AND POD.
      DO 305 K=1,NKP1
      DO 305 J=1,NJP1
```

```
      DO 305 I=1,NIP1
      TOD(I,J,K)=T(I,J,K)
      COD(I,J,K)=C(I,J,K)
      ROD(I,J,K)=R(I,J,K)
      UOD(I,J,K)=U(I,J,K)
      VOD(I,J,K)=V(I,J,K)
      WOD(I,J,K)=W(I,J,K)
      POD(I,J,K)=P(I,J,K)
  305 CONTINUE
C ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C     THIS WRITING IS FOR PLOTTINGS                                      |
C|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
      IF(NTREAL .NE. NTREAL/NTAPE*NTAPE)GOTO 522
CCC   IWRITE=10
CCC   WRITE(IWRITE)
CCC  & TIME,NTREAL,T,R,U,V,W,P,CPM,COND,VIS,QRNET,ITERT,QCORRT,PM1,PM2,
CCC  & H,TA,U0,COND0,VIS0,RHO0,NI,NJ,NK,NIP1,NJP1,NKP1,NIM1,NJM1,NKM1,
CCC  & XC,YC,ZC,XS,YS,ZS,DXXC,DYYC,DZZC,DXXS,DYYS,DZZS
CCC   WRITE(6,*) 'THE TIME WHEN THE DATA WAS STORED ON TAPE IS:',
CCC  & XTIME
C *** ***********************************************************

  522 CONTINUE
C *** ***********************************************************
C     CALL TLEFT(IT)
C     IF(IT.LT.ITLEFT) GO TO 166
C *** ***********************************************************
C TIMREM IS USED TO CALCULATE THE CPU TIME REMAINING AT NPS
      IF (TIMREM(0.).LE.80.) GOTO 166

      GO TO 300
  303 CONTINUE
  277 CONTINUE
      WRITE(6,1111)
 1111 FORMAT(2X,'****** THE MAXIMUM TIME HAS BEEN REACHED ******',I8)
      GO TO 172
C *** ***********************************************************
  166 IF(NTREAL .NE. NTREAL/NTAPE*NTAPE) WRITE(9)
     & TIME,NTREAL,T,R,U,V,W,P,CPM,COND,VIS,QRNET,ITERT,QCORRT,PM1,PM2,
     & H,TA,U0,COND0,VIS0,RHO0,NI,NJ,NK,NIP1,NJP1,NKP1,NIM1,NJM1,NKM1,
     & XC,YC,ZC,XS,YS,ZS,DXXC,DYYC,DZZC,DXXS,DYYS,DZZS
      REWIND 9
C *** ***********************************************************
      GOTO 172
 2020 CONTINUE
      WRITE (6,*) '  RESIDUAL MASS IS LARGER THAN 10.0, PROGRAM STOPS'
  172 CONTINUE
      STOP
      END

C ______________________________________________________________________
* **********************************************************************
      SUBROUTINE INPUT
* **********************************************************************
*     THIS SUBROUTINE SETS UP REQUIRED VALUES TO BEGIN THE PROGRAM.    *
*     VARIABLES ARE:                                                   *
*             KRUN      =      WHEN EQUAL TO ONE,READ FROM THE         *
*                              RESTART DISK, ELSE FROM THE JCL         *
*             NCHIP     =      NUMBER OF SOLID PIECES                  *
*             NWRP      =      NUMBER OF TIME STEPS TO WRITE ON THE    *
*                              PAPER                                   *
*             NTHCO     =      NUMBER OF THERMOCOUPLES TO PRINT OUT    *
*             TMAX      =      MAXIMUM TIME ALLOWED (REAL)             *
*             TWRITE    =      SECONDS IN REAL TIME TO PRINT THE       *
*                              P,V,T FIELDS ON PAPER                   *
*             TTAPE     =      TIME INTERVAL TO WRITE ON THE TAPE      *
```

```
*              DTIME      =      TIME STEP (DIMENSIONLESS)                  *
*              HSZ        =      HEAT SOURCE SIZE, USED TO CALCULATE        *
*                                THE VOLUME OF THE FIRE CELL                *
*              ICHPB      =      FIRST SOLID NODE IN THETA DIRECTION        *
*              JCHPB      =      FIRST SOLID NODE IN R DIRECTION            *
*              KCHPB      =      FIRST SOLID NODE IN PHI DIRECTION          *
*              NCHPI      =      NUMBER OF NODES IN THETA DIRECTION         *
*              NCHPJ      =      NUMBER OF NODES IN R DIRECTION             *
*              NCHPK      =      NUMBER OF NODES IN PHI DIRECTION           *
*              CX,CY,CZ   =      THERMOCOUPLE POSITIONS IN THETA,R,PHI      *
*************************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &         DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &           ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &       ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &      SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)

C #1. READ IN DATA TO INDICATE EITHER KRUN=0 OR 1
      READ(5,*) KRUN,NCHIP,NWRP,NTHCO
C #2. READ IN DATA SET 1 - 6 DATA
      READ(5,*) TMAX,TWRITE,TTAPE,DTIME
C #3. READ IN DATA FOR HEAT SOURCE

      READ (5,*)  HSZ(1,1),HSZ(1,2),HSZ(2,1),HSZ(2,2),HSZ(3,1),HSZ(3,2)
      WRITE(6,20) HSZ(1,1),HSZ(1,2),HSZ(2,1),HSZ(2,2),HSZ(3,1),HSZ(3,2)
   20 FORMAT (/,20X,'HEAT SOURCE LOCATION IS IN THE VOLUME (NON-DIME',
     &   'NSIONAL WITH RESPECT TO RADIUS)',
     &     /,5X,'FROM  ',F8.4,'  TO  ',F8.4,'  IN X-DIRECTION',
     &     /,5X,'FROM  ',F8.4,'  TO  ',F8.4,'  IN Y-DIRECTION',
     &     /,5X,'FROM  ',F8.4,'  TO  ',F8.4,'  IN Z-DIRECTION',/)

C #4. READ IN DECK DATA
      IF (NCHIP.EQ.0) GOTO 16
      PRINT *
      PRINT *,'    THE REGION BOUNDED BY SOLID'
      DO 19 N=1,NCHIP
      READ (5,*) ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
     &          NCHPK(N),TCHP(N),CPS(N),CONS(N)
      WRITE (6,10) N,ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
     &              NCHPK(N),TCHP(N),CPS(N),CONS(N)
   10 FORMAT (2X,'N= ',I2,'  ICHPB= ',I2,' NCHPI= ',I2, '  JCHPB= ',I2,
     &'  NCHPJ=  ',I2, '  KCHPB= ',I2,'  NCHPK= ',I2, ' TCHP= ',F8.5,
     &'  CPS=  ',F8.5, '  CONS =  ',F12.5,/)
   19 CONTINUE
```

```
   16 CONTINUE

C #5.   INPUT THERMOCOUPLE COORDINATE
C       IN TERMS OF X(THETA), Y(RADIUS),Z(PHI)
      PRINT *
      PRINT *, '    THERMOCOUPLE POSITION IN TERMS OF THETA, R, PHI'
      PRINT *
      DO 110 I=1,NTHCO
      READ (5,*) CX(I),CY(I),CZ(I)
      WRITE (6,*) I, CX(I),CY(I),CZ(I)
  110 CONTINUE
      RETURN
      END

C ____________________________________________________________________
C *** ****************************************************************
      SUBROUTINE INIT
C *** ****************************************************************
**********************************************************************
*     THIS SUBROUTINE INITIALIZES THE FIELD AND CONSTANTS WITH RESPECT *
*     TO INITIAL START OR RESTARTING CAPABILITY.                       *
*     VARIABLES ARE :                                                  *
*           TIME        =       DIMENSIONLESS TIME                     *
*           UO          =       CHARACTERISTIC VELOCITY (1 FT/SEC)     *
*           H           =       CHARACTERISTIC LENGTH (RADIUS(9.6FT))  *
*           TR          =       TEMP IN DEGREES KELVIN                 *
*           TA          =       TEMP IN DEGREES RANKINE                *
*           VISO        =       REFERENCE VISCOSITY (NONDIM)           *
*           VISL        =       MINIMUM VISCOSITY (NONDIM)             *
*           VISMAX      =       MAXIMUM VISCOSITY (NONDIM)             *
*           HR          =       RADIUS IN CM                           *
*           CONDO       =       REFERENCE CONDUCTIVITY                 *
*           CO          =       INITIAL SMOKE CONCENTRATION            *
*           NJRA        =       POINT OF RADIATION IN J DIRECTION      *
*                               LOCATED ON THE INNER SOLID BOUNDARY    *
*           HCONV       =       HEAT TRANSFER COEFFICIENT              *
*           HCOEF       =       DIMENSIONLESS HEAT TRANSFER COEF       *
*           CONST1      =       USED TO NONDIMENSIONALIZE PRESSURE     *
*           RHOO        =       REFERENCE DENSITY                      *
*           GC          =       GRAVITY CONSTANT                       *
*           BUOY        =       BUOYANCY FORCE CONSTANT                *
*           UGRT        =       PERFECT GAS LAW NONDIMENSIONAL CONSTANT*
*           CPO         =       REFERENCE SPECIFIC HEAT                *
*           NWRITE/     =       NONDIMENSIONAL FORMS OF TWRITE AND     *
*           NTAPE               TTAPE                                  *
*    MATRICES OF THE FORM                                              *
*           _OD         =       DIMENSIONLESS PARAMETER AT OLD TIME    *
*           _           =       DIMENSIONLESS PARAMETER                *
*           _PD         =       UPDATED DIMENSIONLESS PARAMETER        *
*    WHERE THE PARAMETERS ARE                                          *
*           U,V,W       =       VELOCITY IN THETA, R , PHI DIRECTION   *
*           T,P,C       =       TEMP, PRESSURE, AND SMOKE CONCENTRATION*
*                                                                      *
*           DU,DV,DZ    =       USED IN PRESSURE CORRECTION SUBROUTINE *
*           PP          =       CORRECTED PRESSURE (P')                *
*           SU          =       SOURCE TERM                            *
*           SP          =       TERM AT P NODAL POINT FOR BOUNDARY     *
*                               CONDITIONS                             *
*           AP          =       COEFICIENT AT NODAL POINT              *
*           AE,AW,AN    =       COEFICIENTS AT PTS EAST,WEST,NORTH,    *
*           AS,AF,AB            SOUTH, FRONT, AND BACK                 *
*           SMP         =       RESIDUAL MASS SUMMATION OF NODAL POINT *
*           SMPP        =       LENGTH SCALE FOR TURBULENCE            *
*           CPM         =       MEAN SPECIFIC HEAT                     *
*           VIS         =       VISCOSITY                              *
*           COND        =       CONDUCTIVITY MATRIX                    *
*           NHSZ        =       WHEN THIS VALUE EQUALS ZERO, THERE IS  *
*                               NO HEAT SOURCE LOCATED AT THE NODE     *
```

```
*           NOD             =          IF EQUAL TO ZERO, LIQUID                     *
*                                      IF EQUAL TO ONE, SOLID                       *
*           _B,_E           =          BEGINNING AND ENDING NODAL POINT FOR         *
*                                      THE SOLID IN I,J,K                           *
*           REQ             =          DENSITY AT EQUILIBRIUM                       *
*           NIP1            =          NODAL POINT IN I PLUS 1 (OTHERS SIMILAR)
*           XC,YC,ZC        =          THETA,R,PHI LOCATION OF NODAL POINT OF *
*                                      A CENTER CELL                                *
*           DXXC,DYYC       =          LENGTH AROUND THE CENTER CELL                *
*           DZZC                                                                    *
*           XS,YS,ZS        =          THETA,R,PHI LOCATION OF NODAL POINT OF *
*                                      A STAGGERED CELL                             *
*           DXXS,DYYS       =          LENGTH AROUND THE STAGGERED CELL             *
*           DZZS                                                                    *
*           CX,CY,CZ        =          LOCATION OF THERMOCOUPLE IN THETA,R,PHI*
*****************************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1 PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &          ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &       ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
      DATA GRAV/32.17/
C ***   INTRODUCE GIVEN PARAMETERS
      TIME=0.
      TR=TA/1.8
      H=9.6
      VIS0=VIS0/U0/H
      VISL=VIS0
      VISMAX=400.*VISL
      HR=H*30.48
      COND0=VIS0/PRT
      PI=4.*ATAN(1.)
      ALEW = 1.0
      NJRA=15
C  THE HEAT TRANSFER COEFFICIENT IS IN BTU/HR/FT**2/F
      HCONV=5.0
      HCOEF=HCONV*4./(3600.*CP0*RHO0*U0)
      C0 = 0.0

      CONST1=RHO0*U0*U0/(GC*14.696*144.)
      CONST3=1.8/TA
      CONST4=H*30.48
      CONST6=U0*30.48
      NTMAX0=0
```

```
      BUOY=GRAV*H/(U0*U0)
      UGRT=U0*U0/(GC*RAIR*TA)
      TCOOL=1.0
      CONSRA=TA*TA*TA/(RHO0*CP0*U0*3600.*H*H)*1.714E-9
      WRITE(6,200) TR,COND0,VIS0,CP0,HR,DTIME,HCONV
  200 FORMAT(5X, 'THE REFRENCE TEMPERATURE AND THERMAL PROPERTIES',/,
     &    /,5X, 'T    = ',F10.4,'K,      COND0 = ',E12.6,
     &    /,5X,'VIS0 = ',E12.6,'    CP0   = ',E12.6,
     &    /,5X,'RADIUS = ',E12.6,'   CM',
     &    /,5X,'DTIME = ',E12.6,
     &    /,5X,'HCONV = ',E12.6,/)
      NWRITE=TWRITE*U0/DTIME/H
      NTAPE=TTAPE*U0/DTIME/H
C ***   PRINT OUT INPUT INFORMATION
      WRITE(6,61) (STAR,I=1,90),KRUN,TMAX,TWRITE,TTAPE,NWRP
   61 FORMAT(///,90A1,/,5X,'KRUN  =',I2,/,5X,
     & 'TMAX =',F8.3,' SECONDS',/5X,'TWRITE =',F8.3,
     & ' SECONDS',/,5X,'TTAPE  =',F8.3,' SECONDS',
     & /,5X,' NUMBER INTERVALS OF WRITING ON PAPER  ', I5,/)
C ***   INITIALIZE VARIABLE FIELD
      DO 220 J=1,NJP1
      DO 220 I=1,NIP1
      DO 220 K=1,NKP1
      ROD(I,J,K)=1.
      R(I,J,K)=1.
      RPD(I,J,K)=1.
      UOD(I,J,K)=0.
      U(I,J,K)=0.
      UPD(I,J,K)=0.
      VOD(I,J,K)=0.
      V(I,J,K)=0.
      VPD(I,J,K)=0.
      W(I,J,K)=0.
      WPD(I,J,K)=0.
      WOD(I,J,K)=0.
      POD(I,J,K)=0.
      P(I,J,K)=0.
      PPD(I,J,K)=0.
      DU(I,J,K)=0.
      DV(I,J,K)=0.
      DW(I,J,K)=0.
      SU(I,J,K)=0.
      SP(I,J,K)=0.
      PP(I,J,K)=0.
      AP(I,J,K)=0.
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AN(I,J,K)=0.
      AS(I,J,K)=0.
      AF(I,J,K)=0.
      AB(I,J,K)=0.
      SMP(I,J,K)=0.
      SMPP(I,J,K)=0.
      VIS(I,J,K)=VISL
      COND(I,J,K)=COND0
      CPM(I,J,K)=1.0E0
      TOD(I,J,K)=1.0E0
      T(I,J,K)=TOD(I,J,K)
      TPD(I,J,K)=TOD(I,J,K)
      COD(I,J,K)=C0
      C(I,J,K)=COD(I,J,K)
      CPD(I,J,K)=COD(I,J,K)
      NHSZ(I,J,K)=0
      NOD(I,J,K)=0
  220 CONTINUE
```

```
C ***   DETERMINE THE POSITION OF HEAT SOURCE
        DO 300 I=2,NI
        DO 300 J=2,NJ
C CHANGE TO RECTANGULAR COORDINATES
        XX=YC(J)*COS(XC(I))
        YY=YC(J)*SIN(XC(I))
C CHECK TO SEE IF IN HS CONTROL VOLUME, IF SO SET NHSZ=1
        IF (XX.LT.HSZ(1,1).OR.XX.GT.HSZ(1,2)) GOTO 310
        IF (YY.LT.HSZ(2,1).OR.YY.GT.HSZ(2,2)) GOTO 310
        NHSZ(I,J,16)=1
        NHSZ(I,J,17)=1
  315 FORMAT (2X,10(4X,I4,2X,I4))
        GOTO 300
  310 CONTINUE
  300 CONTINUE

C ***     DEFINE THERMAL PROPERTIES OF DECK AND SOLID
        IF (NCHIP.EQ.0) GOTO 410
        DO 402 N=1,NCHIP
        IB=ICHPB(N)
        IE=IB+NCHPI(N)-1
        JB=JCHPB(N)
        JE=JB+NCHPJ(N)-1
        KB=KCHPB(N)
        KE=KB+NCHPK(N)-1
        DO 405 I=IB,IE-1
        DO 405 J=JB,JE-1
        DO 405 K=KB,KE-1
        COND(I,J,K)=COND0*CONS(N)
        CPM(I,J,K)=CP0*CPS(N)
        NOD(I,J,K)=1
  405 CONTINUE
  402 CONTINUE
  410 CONTINUE


C *** FOR CONTINUING RUN, READ DATA FROM TAPE OR DISK
        IF(KRUN .EQ. 1) GO TO 9997
        GO TO 15
 9997 READ(8,END=9998)
     & TIME,NTMAX0,TOD,ROD,UOD,VOD,WOD,POD,CPM,COND,VIS,QRNET,ITERT,QCOR
     &RT,PM1,PM2,XX,XX,XX,XX,XX,XX,NI,NJ,NK,NIP1,NJP1,NKP1,NIM1,NJM1
     & ,NKM1,XC,YC,ZC,XS,YS,ZS,DXXC,DYYC,DZZC,DXXS,DYYS,DZZS
        GO TO 9997
 9998 CONTINUE
        REWIND 8
        CLOSE (8)
        WRITE(6,*)NTMAX0
   15 CONTINUE

C ***    DEFINE HEIGHT OF NODE POINTS AND COMPUTE HYDROSTATIC
C        EQUILIBRIUM DENSITY REQ(I,J,K)

        DO 13 K=1,NKP1
        DO 13 I=1,NIP1
        DO 13 J=1,NJP1
        DHY=YC(J)*SIN(XC(I))*SIN(ZC(K))
        HEIGHT(I,J,K)=DHY
 13   CONTINUE
C
        DO 229 J=1,NJP1
        DO 229 I=1,NIP1
        DO 229 K=1,NKP1
        AAAA=-BUOY*UGRT*HEIGHT(I,J,K)
```

```
      REQ(I,J,K)=EXP(AAAA)
      IF(KRUN .NE. 0) GO TO 229
      RPD(I,J,K)=REQ(I,J,K)/TPD(I,J,K)
      ROD(I,J,K)=RPD(I,J,K)
      R(I,J,K)=RPD(I,J,K)
  229 CONTINUE
C ***    INITIALIZE U,V,T,R,P FIELD
      DO 210 K=1,NKP1
      DO 210 J=1,NJP1
      DO 210 I=1,NIP1
      T(I,J,K)=TOD(I,J,K)
      C(I,J,K)=COD(I,J,K)
      R(I,J,K)=ROD(I,J,K)
      U(I,J,K)=UOD(I,J,K)
      V(I,J,K)=VOD(I,J,K)
      W(I,J,K)=WOD(I,J,K)
      P(I,J,K)=POD(I,J,K)
  210 CONTINUE
C ***   FOLLOWING IS FOR DETERMINING THE THERMOCOUPLE POSITIONS
      DO 5000 N=1,NTHCO
      DO 5001 I=1,NIP1
      IF (XC(I).LT.CX(N).AND.XC(I+1).GE.CX(N)) GOTO 5002
 5001 CONTINUE
 5002 II=I
      DO 5003 J=1,NJP1
      IF (YC(J).LT.CY(N).AND.YC(J+1).GE.CY(N)) GOTO 5004
 5003 CONTINUE
 5004 JJ=J
      DO 5005 K=1,NKP1
      IF (ZC(K).LT.CZ(N).AND.ZC(K+1).GE.CZ(N)) GOTO 5006
 5005 CONTINUE
 5006 KK=K
      NTH(N,1)=II
      NTH(N,2)=JJ
      NTH(N,3)=KK
 5000 CONTINUE

      RETURN
      END
C
C *** ****************************************************************
      SUBROUTINE CALVIS
C *** ****************************************************************
*     THIS SUBROUTINE CALCULATES THE TURBULENT VISCOSITY AND UPDATES*
*     THE VISCOSITY MATRIX                                          *
********************************************************************

      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CPO,PRT,CONDO,VISO,RHOO,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &        AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
```

```
C ***    CALCULATE LOCAL SHEAR AND VISCOSITY VIS(I,J,K)
C
C ***    SPECIFY LOCAL TURBULENT LENGTH SCALES  SMPP(I,J,K)
      DO 611 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 611 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 611 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2)  IM2=NIM1
      IF (I.EQ.NI) IP2=3
      IF (NOD(I,J,K).EQ.1) GOTO 611
C        CENTRAL LENGTH OF THE SCALE CONTROL VOLUME
      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)
      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)
      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)
      IF (J.EQ.2) DYS=DYS/2.
      IF (K.EQ.2) DZB=DZB/2.
      IF (J.NE.NJ) GOTO 101
      JP2=JP1
      DYN=DYN/2.
  101 IF (K.NE.NK) GOTO 102
      KP2=KP1
      DZF=DZF/2.
  102 CONTINUE
C ***    CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T
      DXE =XL(IP1,J,K,0,1)
      DXW =XL(I  ,J,K,0,1)
      DYN =YL(I,JP1,K,0,2)
      DYS =YL(I,J  ,K,0,2)
      DZF =ZL(I,J,KP1,0,3)
      DZB =ZL(I,J,K  ,0,3)
C ***    CACULATE DV/DX,D2V/DX2,DU/DX,D2U/DX2,DW/DX AND D2W/DX2

      DUDX=(U(IP1,J,K)-U(I,J,K))/DXI
      DUDXW=0.5*(U(IP1,J,K)-U(IM1,J,K))/DXW
      DUDXE=0.5*(U(IP2,J,K)-U(I  ,J,K))/DXE
      D2UDX2=(DUDXE-DUDXW)/DXI

      DVDXW=0.5*(V(I,JP1,K)+V(I,J,K)-V(IM1,JP1,K)-V(IM1,J,K))/DXW
      DVDXE=0.5*(V(IP1,JP1,K)+V(IP1,J,K)-V(I,JP1,K)-V(I,J,K))/DXE
      DVDX=0.5*(DVDXE+DVDXW)
      D2VDX2=  (DVDXE-DVDXW)/DXI

      DWDXW=0.5*(W(I,J,KP1)+W(I,J,K)-W(IM1,J,KP1)-W(IM1,J,K))/DXW
      DWDXE=0.5*(W(IP1,J,KP1)+W(IP1,J,K)-W(I,J,KP1)-W(I,J,K))/DXE
      DWDX=0.5*(DWDXE+DWDXW)
      D2WDX2=  (DWDXE-DWDXW)/DXI
```

```
  602 CONTINUE
C ***    CALCULATE DU/DY,D2U/DY2,DV/DY,D2V/DY2,DW/DY AND D2W/DY2

      DVDY=(V(I,JP1,K)-V(I,J,K))/DYJ
      DVDYS=0.5*(V(I,JP1,K)-V(I,JM1,K))/DYS
      DVDYN=0.5*(V(I,JP2,K)-V(I,J  ,K))/DYN
      D2VDY2=(DVDYN-DVDYS)/DYJ

      DUDYS=0.5*(U(IP1,J,K)+U(I,J,K)-U(IP1,JM1,K)-U(I,JM1,K))/DYS
      DUDYN=0.5*(U(IP1,JP1,K)+U(I,JP1,K)-U(IP1,J,K)-U(I,J,K))/DYN
      DUDY=0.5*(DUDYN+DUDYS)
      D2UDY2=  (DUDYN-DUDYS)/DYJ

      DWDYS=0.5*(W(I,J,KP1)+W(I,J,K)-W(I,JM1,KP1)-W(I,JM1,K))/DYS
      DWDYN=0.5*(W(I,JP1,KP1)+W(I,JP1,K)-W(I,J,KP1)-W(I,J,K))/DYN
      DWDY=0.5*(DWDYN+DWDYS)
      D2WDY2=  (DWDYN-DWDYS)/DYJ
  606 CONTINUE
C ***    CALCULATE DU/DZ,D2U/DZ2,DV/DZ,D2V/DZ2,DW/DZ AND D2W/DZ2

      DWDZ=(W(I,J,KP1)-W(I,J,K))/DZK
      DWDZF=0.5*(W(I,J,KP2)-W(I,J,K  ))/DZF
      DWDZB=0.5*(W(I,J,KP1)-W(I,J,KM1))/DZB
      D2WDZ2=(DWDZF-DWDZB)/DZK

      DVDZB=0.5*(V(I,JP1,K)+V(I,J,K)-V(I,JP1,KM1)-V(I,J,KM1))/DZB
      DVDZF=0.5*(V(I,JP1,KP1)+V(I,J,KP1)-V(I,JP1,K)-V(I,J,K))/DZF
      DVDZ=0.5*(DVDZF+DVDZB)
      D2VDZ2=  (DVDZF-DVDZB)/DZK

      DUDZB=0.5*(U(IP1,J,K)+U(I,J,K)-U(IP1,J,KM1)-W(I,J,KM1))/DZB
      DUDZF=0.5*(U(IP1,J,KP1)+U(I,J,KP1)-U(IP1,J,K)-U(I,J,K))/DZF
      DUDZ=0.5*(DUDZF+DUDZB)
      D2UDZ2=  (DUDZF-DUDZB)/DZK                                        0108400
      DRDX=((R(IP1,J,K)-REQ(IP1,J,K))-(R(IM1,J,K)-REQ(IM1,J,K)))/
     &    (DXE+DXW)
      DRDY=((R(I,JP1,K)-REQ(I,JP1,K))-(R(I,JM1,K)-REQ(I,JM1,K)))/
     &    (DYN+DYS)
      DRDZ=((R(I,J,KP1)-REQ(I,J,KP1))-(R(I,J,KM1)-REQ(I,J,KM1)))/
     &    (DZF+DZB)
      DRDGA=SIN(ZC(K))*SIN(XC(I))*DRDY+COS(XC(I))*DRDX
     &     +COS(ZC(K))*SIN(XC(I))*DRDZ
C ***    CALCULATE RICHARDSON NUMBER
      STRAIN=DUDY**2+DVDX**2+DWDX**2+DVDZ**2+DWDY**2+DUDZ**2
      DDO2 = SQRT(DUDY*DUDY+DUDX*DUDX+DUDZ*DUDZ+DVDY*DVDY+DVDX*DVDX+
     & DVDZ*DVDZ+DWDX*DWDX+DWDY*DWDY+DWDZ*DWDZ)
      IF(DDO2.EQ.0.)GO TO 600
C ***    CALCULATE TURBULENT LENGTH SCALE SMPP(I,J)
      SMP123=SQRT(U(I,J,K)*U(I,J,K)+V(I,J,K)*V(I,J,K)+W(I,J,K)*W(I,J,K))
     & /DDO2
      SMPP12=DDO2 /SQRT(D2UDX2*D2UDX2+D2UDY2*D2UDY2
     & +D2UDZ2*D2UDZ2+D2VDX2*D2VDX2+D2VDY2*D2VDY2+D2VDZ2*D2VDZ2+
     & D2WDZ2*D2WDZ2+D2WDX2*D2WDX2+D2WDY2*D2WDY2)
      SMPP(I,J,K)=CNT*(SMP123+SMPP12)*.5
      RI(I,J,K)=-BUOY*DRDGA/(R(I,J,K)*STRAIN)
      ABRIPR=ABTURB+RI(I,J,K)/PRT
      IF(ABRIPR .LT. 0.) GO TO 600
      IF(ABRIPR .EQ. 0.) GO TO 613
      GO TO 610
  600 VIS(I,J,K)=VISL
      GO TO 611
```

```
613 VIS(I,J,K)=VISMAX
     GO TO 611
610 VIS(I,J,K)=VISL+R(I,J,K)*SMPP(I,J,K)*SMPP(I,J,K)*SQRT(STRAIN)/
    &          (BTURB*ABRIPR)
     IF(VIS(I,J,K) .GT. VISMAX) VIS(I,J,K)=VISMAX
611 CONTINUE

     DO 110 I=1,NIP1
     DO 110 J=1,NJP1
     VIS(I,J,NKP1)=VIS(I,J,NK)
     VIS(I,J,1   )=VIS(I,J,2 )
110 CONTINUE
     DO 120 J=1,NJP1
     DO 120 K=1,NKP1
     VIS(NIP1,J,K)=VIS(2,J,K)
     VIS(1   ,J,K)=VIS(NI,J,K)
120 CONTINUE
     DO 130 K=1,NKP1
     DO 130 I=1,NIP1
     VIS(I,NJP1,K)=VIS(I,NJ,K)
     VIS(I,1   ,K)=VIS(I,2 ,K)
130 CONTINUE
     DO 140 I=1,NIP1
     DO 140 J=1,NJP1
     DO 140 K=1,NKP1
     IF (NOD(I,J,K).EQ.1) GOTO 140
     COND(I,J,K)=VIS(I,J,K)/PRT
140 CONTINUE
     RETURN
     END


C ____________________________________________________________________
C *** ****************************************************************
     SUBROUTINE CALT
C *** ****************************************************************
     COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
    &         DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
     COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
     COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
    &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
     COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
     COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
     COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
    &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
     COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
    &            NCHPK(10),TCHP(10),CPS(10),CONS(10)
     COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
    &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
     COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
    &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
     COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
    &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
     COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
    &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
    &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
     COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
    &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
    &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
     COMMON/BL37/VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
    &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
C ***    CALCULATE COEFFICIENTS
     DO 100 K=2,NK
     KP2=K+2
     KP1=K+1
     KM1=K-1
```

```
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2)  IM2=NIM1
      IF (I.EQ.NI) IP2=3
C       CENTRAL LENGTH OF THE TEMPERTURE CONTROL VOLUME
      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)
      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)
      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)
C ***    SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)
      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)
      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)
C ***    CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T
      DXEE=XL(IP2,J,K,0,1)
      DXE =XL(IP1,J,K,0,1)
      DXW =XL(I  ,J,K,0,1)
      DXWW=XL(IM1,J,K,0,1)
      DYNN=YL(I,JP2,K,0,2)
      DYN =YL(I,JP1,K,0,2)
      DYS =YL(I,J  ,K,0,2)
      DYSS=YL(I,JM1,K,0,2)
      DZFF=ZL(I,J,KP2,0,3)
      DZF =ZL(I,J,KP1,0,3)
      DZB =ZL(I,J,K  ,0,3)
      DZBB=ZL(I,J,KM1,0,3)
C ***   DEFINE THE AREA OF THE CONTROL VOLUME
      DXYF=DXF*DYF
      DXYB=DXB*DYB
      DYZE=DYE*DZE
      DYZW=DYW*DZW
      DZXN=DZN*DXN
      DZXS=DZS*DXS
      VOL=DXI*DYJ*DZK
      VOLDT=VOL/DTIME
      ZXOYN=DZXN/DYN
      ZXOYS=DZXS/DYS
      XYOZF=DXYF/DZF
      XYOZB=DXYB/DZB
```

```
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW
      GN=(R(I,J,K)*DYP1+R(I,JP1,K)*DYJ)/(DYP1+DYJ)
      GS=(R(I,J,K)*DYM1+R(I,JM1,K)*DYJ)/(DYM1+DYJ)
      GE=(R(I,J,K)*DXP1+R(IP1,J,K)*DXI)/(DXP1+DXI)
      GW=(R(I,J,K)*DXM1+R(IM1,J,K)*DXI)/(DXM1+DXI)
      GF=(R(I,J,K)*DZP1+R(I,J,KP1)*DZK)/(DZP1+DZK)
      GB=(R(I,J,K)*DZM1+R(I,J,KM1)*DZK)/(DZM1+DZK)
      CN=GN*V(I,JP1,K)*DZXN
      CS=GS*V(I,J  ,K)*DZXS
      CE=GE*U(IP1,J,K)*DYZE
      CW=GW*U(I  ,J,K)*DYZW
      CF=GF*W(I,J,KP1)*DXYF
      CB=GB*W(I,J,K  )*DXYB

      CONDN=1./((1./COND(I,J,K)*DYJ+1./COND(I,JP1,K)*DYP1)/(DYP1+DYJ))
      CONDS=1./((1./COND(I,J,K)*DYJ+1./COND(I,JM1,K)*DYM1)/(DYM1+DYJ))
      CONDE=1./((1./COND(I,J,K)*DXI+1./COND(IP1,J,K)*DXP1)/(DXP1+DXI))
      CONDW=1./((1./COND(I,J,K)*DXI+1./COND(IM1,J,K)*DXM1)/(DXM1+DXI))
      CONDF=1./((1./COND(I,J,K)*DZK+1./COND(I,J,KP1)*DZP1)/(DZP1+DZK))
      CONDB=1./((1./COND(I,J,K)*DZK+1./COND(I,J,KM1)*DZM1)/(DZM1+DZK))
      CONDN1=ZXOYN*CONDN
      CONDS1=ZXOYS*CONDS
      CONDE1=YZOXE*CONDE
      CONDW1=YZOXW*CONDW
      CONDF1=XYOZF*CONDF
      CONDB1=XYOZB*CONDB
      CEP=(ABS(CE)+CE)*DXE/DXI/16.
      CEM=(ABS(CE)-CE)*DXE/DXP1/16.
      CWP=(ABS(CW)+CW)*DXW/DXM1/16.
      CWM=(ABS(CW)-CW)*DXW/DXI/16.
      CNP=(ABS(CN)+CN)*DYN/DYJ/16.
      CNM=(ABS(CN)-CN)*DYN/DYP1/16.
      CSP=(ABS(CS)+CS)*DYS/DYM1/16.
      CSM=(ABS(CS)-CS)*DYS/DYJ/16.
      CFP=(ABS(CF)+CF)*DZF/DZK/16.
      CFM=(ABS(CF)-CF)*DZF/DZP1/16.
      CBP=(ABS(CB)+CB)*DZB/DZM1/16.
      CBM=(ABS(CB)-CB)*DZB/DZK/16.
      AE(I,J,K)=-.5*CE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE
      AW(I,J,K)= .5*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW
      AN(I,J,K)=-.5*CN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN
      AS(I,J,K)= .5*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS
      AF(I,J,K)=-.5*CF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF
      AB(I,J,K)= .5*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB
801   AEE=-CEM*DXE/DXEE
      AEER=AEE*TPD(IP2,J,K)*CPM(IP2,J,K)
802   CONTINUE
803   AWW=-CWP*DXW/DXWW
      AWWR=AWW*TPD(IM2,J,K)*CPM(IM2,J,K)
804   CONTINUE
      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
805   ANN=-CNM*DYN/DYNN
      ANNR=ANN*TPD(I,JP2,K)*CPM(I,JP2,K)
806   CONTINUE
      IF (J.GT.2) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
807   ASS=-CSP*DYS/DYSS
      ASSR=ASS*TPD(I,JM2,K)*CPM(I,JM2,K)
```

```
  808 CONTINUE
      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*TPD(I,J,KP2)*CPM(I,J,KP2)
  810 CONTINUE
      IF (K.GT.2) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*TPD(I,J,KM2)*CPM(I,J,KM2)
  812 CONTINUE


C ##########################################################
C ##########################################################
C *** MODIFICATION FOR DECK    BOUNDARIES
  900 CONTINUE
      IF (NOD(IM1,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0
  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
      AEE=0.0
      AEER=0.0
  902 CONTINUE
      IF (NOD(I,JM1,K).EQ.0) GOTO 903
      ASS=0.0
      ASSR=0.0
  903 CONTINUE
      IF (NOD(I,JP1,K).EQ.0) GOTO 904
      ANN=0.0
      ANNR=0.0
  904 CONTINUE
      IF (NOD(I,J,KM1).EQ.0) GOTO 905
      ABB=0.0
      ABBR=0.0
  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0
  906 CONTINUE
C ############################################################
C ############################################################
      AP(I,J,K)=(AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)
     &          +AF(I,J,K)+AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB)*CPM(I,J,K)
     &          +CONDE1+CONDW1+CONDN1+CONDS1+CONDF1+CONDB1
      AE(I,J,K)=AE(I,J,K)*CPM(IP1,J,K)+CONDE1
      AW(I,J,K)=AW(I,J,K)*CPM(IM1,J,K)+CONDW1
      AN(I,J,K)=AN(I,J,K)*CPM(I,JP1,K)+CONDN1
      AS(I,J,K)=AS(I,J,K)*CPM(I,JM1,K)+CONDS1
      AF(I,J,K)=AF(I,J,K)*CPM(I,J,KP1)+CONDF1
      AB(I,J,K)=AB(I,J,K)*CPM(I,J,KM1)+CONDB1
      SP(I,J,K)=-ROD(I,J,K)*VOLDT*CPM(I,J,K)
      SU(I,J,K)= ROD(I,J,K)*VOLDT*TOD(I,J,K)*CPM(I,J,K)
      SU(I,J,K)=SU(I,J,K)+AEER+AWWR+ANNR+ASSR+AFFR+ABBR
  100 CONTINUE
C ***    TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU
C ***    RADIUS DIRECTION
```

```
      DO 500 I=2,NI
      DO 500 K=2,NK
CC    SP(I,2,K)=SP(I,2,K)+AS(I,2,K)
      SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
      SU(I,2,K)=SU(I,2,K)+2.0*AS(I,2,K)*TPD(I,1,K)
      SP(I,NJ,K)=SP(I,NJ,K)-AN(I,NJ,K)
      SU(I,NJ,K)=SU(I,NJ,K)+2.*TPD(I,NJP1,K)*AN(I,NJ,K)
      AS(I,2,K)=0.
      AN(I,NJ,K)=0.
  500 CONTINUE
C ***    CYLIC CONDITIONS
      DO 600 J=2,NJ
      DO 600 K=2,NK
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*T(1    ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*T(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
  600 CONTINUE
C ***      END OF SPHERE
      DO 700 I=2,NI
      DO 700 J=2,NJ
      SP(I,J,2)=SP(I,J,2)+AB(I,J,2)
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)
      AB(I,J,2)=0.
      AF(I,J,NK)=0.
 700  CONTINUE

C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
      DO 300 K=2,NK
      DO 300 J=2,NJ
      DO 300 I=2,NI
      AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
  300 CONTINUE


C *** VOLUME HEAT SOURCE INPUT
      VOLT=0.0
      DO 113 I=2,NI
      DO 113 J=2,NJ
      DO 113 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 113
      DXI =XL(I   ,J,K,0,0)
      DYJ =YL(I,J   ,K,0,0)
      DZK =ZL(I,J,K   ,0,0)
      VOL=DXI*DYJ*DZK*H*H*H
      VOLT=VOLT+VOL
  113 CONTINUE
      DO 111 I=2,NI
      DO 111 J=2,NJ
      DO 111 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 111
      DXI =XL(I   ,J,K,0,0)
      DYJ =YL(I,J   ,K,0,0)
      DZK =ZL(I,J,K   ,0,0)
      QQQ=Q*H/(U0*CP0*RHO0*TA)
      VOL=DXI*DYJ*DZK
      SU(I,J,K)=SU(I,J,K)+VOL*QQQ/VOLT
  111 CONTINUE

C ***   RADIATION INTO THE WALL
      DO 310 K=3,NKM1
      DO 310 I=2,NI
      II=(K-3)*(NI-1)+I-1
```

```
      SU(I,NJRA,K)=SU(I,NJRA,K)-RWALL(II)
  310 CONTINUE
C ***   END OF RADIATION
C ***   SOLVE FOR T
      CALL TRID (2,2,2,NI,NJ,NK,T)
C **** RESET TEMPERATURE AT R=0.0 AND END OF SPHERE
      DO 81 K=1,NKP1
      AVT=0.0
      DO 82 I=2,NI
      AVT=AVT+(T(I,2,K)/NIM1)
   82 CONTINUE
      DO 83 I=1,NIP1
      T(I,1,K)=AVT
   83 CONTINUE
   81 CONTINUE
C
      DO 74 I=1,NIP1
      DO 74 J=1,NJP1
      T(I,J,1)=T(I,J,2)
      T(I,J,NKP1)=T(I,J,NK)
   74 CONTINUE
C ***    FOR SURFACE HEAT EXCHANGE WITH SURROUNDING
      DO 84 I=2,NI
      DO 84 K=2,NK
      DYJ=YL(I,NJ,K,0,0)
      T(I,NJP1,K)=(2.0*COND(I,NJ,K)*T(I,NJ,K)/DYJ+HCOEF*TINF)/
     &            (HCOEF+2.0*COND(I,NJ,K)/DYJ)
CC    T(I,NJP1,K)=T(I,NJM1,K)
CC    T(I,NJ   ,K)=T(I,NJM1,K)
  84  CONTINUE

C ***        FOR CYLIC CONDITION
      DO 80 J=1,NJP1
      DO 80 K=1,NKP1
      T(1,J,K)=T(NI,J,K)
      T(NIP1,J,K)=T(2,J,K)
  80  CONTINUE
      RETURN
      END


C ____________________________________________________________________
C *** ****************************************************************
      SUBROUTINE CALC
C *** ****************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,UO,H,UGRT,BUOY,
     &  CPO,PRT,CONDO,VISO,RHOO,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &            NCHPK(10),TCHP(10),CPS(10),CONS(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
```

```
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &        AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &      ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
C ***    CALCULATE COEFFICIENTS
      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2)  IM2=NIM1
      IF (I.EQ.NI) IP2=3
C       CENTRAL LENGTH OF THE SCALE CONTROL VOLUME
      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)
      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)
      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)
C ***   SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)
      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)
      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)
C ***   CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T
      DXEE=XL(IP2,J,K,0,1)
      DXE =XL(IP1,J,K,0,1)
      DXW =XL(I  ,J,K,0,1)
      DXWW=XL(IM1,J,K,0,1)
      DYNN=YL(I,JP2,K,0,2)
      DYN =YL(I,JP1,K,0,2)
      DYS =YL(I,J  ,K,0,2)
      DYSS=YL(I,JM1,K,0,2)
      DZFF=ZL(I,J,KP2,0,3)
      DZF =ZL(I,J,KP1,0,3)
      DZB =ZL(I,J,K  ,0,3)
      DZBB=ZL(I,J,KM1,0,3)
C ***   DEFINE THE AREA OF THE CONTROL VOLUME
```

```
      DXYF=DXF*DYF
      DXYB=DXB*DYB
      DYZE=DYE*DZE
      DYZW=DYW*DZW
      DZXN=DZN*DXN
      DZXS=DZS*DXS
      VOL=DXI*DYJ*DZK
      VOLDT=VOL/DTIME
      ZXOYN=DZXN/DYN
      ZXOYS=DZXS/DYS
      XYOZF=DXYF/DZF
      XYOZB=DXYB/DZB
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW
      GN=(R(I,J,K)*DYP1+R(I,JP1,K)*DYJ)/(DYP1+DYJ)
      GS=(R(I,J,K)*DYM1+R(I,JM1,K)*DYJ)/(DYM1+DYJ)
      GE=(R(I,J,K)*DXP1+R(IP1,J,K)*DXI)/(DXP1+DXI)
      GW=(R(I,J,K)*DXM1+R(IM1,J,K)*DXI)/(DXM1+DXI)
      GF=(R(I,J,K)*DZP1+R(I,J,KP1)*DZK)/(DZP1+DZK)
      GB=(R(I,J,K)*DZM1+R(I,J,KM1)*DZK)/(DZM1+DZK)
      CN=GN*V(I,JP1,K)*DZXN
      CS=GS*V(I,J  ,K)*DZXS
      CE=GE*U(IP1,J,K)*DYZE
      CW=GW*U(I  ,J,K)*DYZW
      CF=GF*W(I,J,KP1)*DXYF
      CB=GB*W(I,J,K  )*DXYB

      CONDN=1./((1./COND(I,J,K)*DYJ+1./COND(I,JP1,K)*DYP1)/(DYP1+DYJ))
      CONDS=1./((1./COND(I,J,K)*DYJ+1./COND(I,JM1,K)*DYM1)/(DYM1+DYJ))
      CONDE=1./((1./COND(I,J,K)*DXI+1./COND(IP1,J,K)*DXP1)/(DXP1+DXI))
      CONDW=1./((1./COND(I,J,K)*DXI+1./COND(IM1,J,K)*DXM1)/(DXM1+DXI))
      CONDF=1./((1./COND(I,J,K)*DZK+1./COND(I,J,KP1)*DZP1)/(DZP1+DZK))
      CONDB=1./((1./COND(I,J,K)*DZK+1./COND(I,J,KM1)*DZM1)/(DZM1+DZK))
      CONDN1=ZXOYN*CONDN*ALEW
      CONDS1=ZXOYS*CONDS*ALEW
      CONDE1=YZOXE*CONDE*ALEW
      CONDW1=YZOXW*CONDW*ALEW
      CONDF1=XYOZF*CONDF*ALEW
      CONDB1=XYOZB*CONDB*ALEW
      CEP=(ABS(CE)+CE)*DXE/DXI/16.
      CEM=(ABS(CE)-CE)*DXE/DXP1/16.
      CWP=(ABS(CW)+CW)*DXW/DXM1/16.
      CWM=(ABS(CW)-CW)*DXW/DXI/16.
      CNP=(ABS(CN)+CN)*DYN/DYJ/16.
      CNM=(ABS(CN)-CN)*DYN/DYP1/16.
      CSP=(ABS(CS)+CS)*DYS/DYM1/16.
      CSM=(ABS(CS)-CS)*DYS/DYJ/16.
      CFP=(ABS(CF)+CF)*DZF/DZK/16.
      CFM=(ABS(CF)-CF)*DZF/DZP1/16.
      CBP=(ABS(CB)+CB)*DZB/DZM1/16.
      CBM=(ABS(CB)-CB)*DZB/DZK/16.
      AE(I,J,K)=-.5*CE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE
      AW(I,J,K)= .5*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW
      AN(I,J,K)=-.5*CN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN
      AS(I,J,K)= .5*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS
      AF(I,J,K)=-.5*CF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF
      AB(I,J,K)= .5*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB
  801 AEE=-CEM*DXE/DXEE
      AEER=AEE*CPD(IP2,J,K)
  802 CONTINUE
  803 AWW=-CWP*DXW/DXWW
      AWWR=AWW*CPD(IM2,J,K)
  804 CONTINUE
```

```
      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
  805 ANN=-CNM*DYN/DYNN
      ANNR=ANN*CPD(I,JP2,K)
  806 CONTINUE
      IF (J.GT.2) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
  807 ASS=-CSP*DYS/DYSS
      ASSR=ASS*CPD(I,JM2,K)
  808 CONTINUE
      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*CPD(I,J,KP2)
  810 CONTINUE
      IF (K.GT.2) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*CPD(I,J,KM2)
  812 CONTINUE


C ############################################################
C ############################################################
C *** MODIFICATION FOR DECK    BOUNDARIES
  900 CONTINUE
      IF (NOD(IM1,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0
  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
      AEE=0.0
      AEER=0.0
  902 CONTINUE
      IF (NOD(I,JM1,K).EQ.0) GOTO 903
      ASS=0.0
      ASSR=0.0
  903 CONTINUE
      IF (NOD(I,JP1,K).EQ.0) GOTO 904
      ANN=0.0
      ANNR=0.0
  904 CONTINUE
      IF (NOD(I,J,KM1).EQ.0) GOTO 905
      ABB=0.0
      ABBR=0.0
  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0
  906 CONTINUE
C ############################################################
C ############################################################
      AP(I,J,K)=(AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)
     &          +AF(I,J,K)+AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB)
     &          +CONDE1+CONDW1+CONDN1+CONDS1+CONDF1+CONDB1
```

```
      AE(I,J,K)=AE(I,J,K)+CONDE1
      AW(I,J,K)=AW(I,J,K)+CONDW1
      AN(I,J,K)=AN(I,J,K)+CONDN1
      AS(I,J,K)=AS(I,J,K)+CONDS1
      AF(I,J,K)=AF(I,J,K)+CONDF1
      AB(I,J,K)=AB(I,J,K)+CONDB1
      SP(I,J,K)=-ROD(I,J,K)*VOLDT
      SU(I,J,K)= ROD(I,J,K)*VOLDT*TOD(I,J,K)
      SU(I,J,K)=SU(I,J,K)+AEER+AWWR+ANNR+ASSR+AFFR+ABBR
  100 CONTINUE
C ***    TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU
C
C ***    RADIUS DIRECTION
      DO 500 I=2,NI
      DO 500 K=2,NK
CC     SP(I,2,K)=SP(I,2,K)+AS(I,2,K)
      SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
      SU(I,2,K)=SU(I,2,K)+2.0*AS(I,2,K)*CPD(I,1,K)
      SP(I,NJ,K)=SP(I,NJ,K)-AN(I,NJ,K)
      SU(I,NJ,K)=SU(I,NJ,K)+2.*CPD(I,NJP1,K)*AN(I,NJ,K)
      AS(I,2,K)=0.
      AN(I,NJ,K)=0.
  500 CONTINUE
C ***    CYLIC CONDITIONS
      DO 600 J=2,NJ
      DO 600 K=2,NK
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*C(1   ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*C(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
  600 CONTINUE
C ***      END OF SPHERE
      DO 700 I=2,NI
      DO 700 J=2,NJ
      SP(I,J,2)=SP(I,J,2)+AB(I,J,2)
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)
      AB(I,J,2)=0.
      AF(I,J,NK)=0.
 700  CONTINUE


C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
      DO 300 K=2,NK
      DO 300 J=2,NJ
      DO 300 I=2,NI
      AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
  300 CONTINUE


C *** VOLUME MASS SOURCE INPUT
      VOLT=0.0
      DO 113 I=2,NI
      DO 113 J=2,NJ
      DO 113 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 113
      DXI =XL(I  ,J,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DZK =ZL(I,J,K  ,0,0)
      VOL=DXI*DYJ*DZK*H*H*H
      VOLT=VOLT+VOL
  113 CONTINUE
      DO 111 I=2,NI
      DO 111 J=2,NJ
```

```
      DO 111 K=16,17
      IF (NHSZ(I,J,K).EQ.0) GOTO 111
      DXI =XL(I  ,J,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DZK =ZL(I,J,K  ,0,0)
      QQQ=Q*H/(U0*CP0*RHO0*TA)
      QMS= 1.0
      QMS = QMS*H/(U0*RHO0)
      VOL=DXI*DYJ*DZK
      SU(I,J,K)=SU(I,J,K)+VOL*QMS/VOLT
  111 CONTINUE
C ***   SOLVE FOR C
      CALL TRID (2,2,2,NI,NJM1,NK,C)
C **** RESET CONCENTRATION AT R=0.0 AND END OF SPHERE
      DO 81 K=1,NKP1
      AVT=0.0
      DO 82 I=2,NI
      AVT=AVT+(C(I,2,K)/NIM1)
   82 CONTINUE
      DO 83 I=1,NIP1
      C(I,1,K)=AVT
   83 CONTINUE
   81 CONTINUE
      DO 74 I=1,NIP1
      DO 74 J=1,NJP1
      C(I,J,1)=C(I,J,2)
      C(I,J,NKP1)=C(I,J,NK)
   74 CONTINUE
C ***    FOR SURFACE MASS EXCHANGE WITH SURROUNDING
      DO 84 I=2,NI
      DO 84 K=2,NK
      C(I,NJP1,K)=C(I,NJ,K)
  84  CONTINUE

C ***        FOR CYLIC CONDITION
      DO 80 J=1,NJP1
      DO 80 K=1,NKP1
      C(1,J,K)=C(NI,J,K)
      C(NIP1,J,K)=C(2,J,K)
  80  CONTINUE
      RETURN
      END

C
C    *****************************************************************
      SUBROUTINE CALU
C    *****************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &            ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &             NCHPK(10),TCHP(10),CPS(10),CONS(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
```

```
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &      ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &      SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &    DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &    SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &      ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
C ***   CALCULATE COEFFICIENTS
      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2) IM1=NI
      IF (I.EQ.2) IM2=NIM1
      IF (I.EQ.3) IM2=NI
      IF (I.EQ.NI) IP2=3

C       CENTRAL LENGTH OF THE SCALE CONTROL VOLUME
      DXP1=XL(IP1,J,K,1,0)
      DXI =XL(I  ,J,K,1,0)
      DXM1=XL(IM1,J,K,1,0)
      DYP1=YL(I,JP1,K,1,0)
      DYJ =YL(I,J  ,K,1,0)
      DYM1=YL(I,JM1,K,1,0)
      DZP1=ZL(I,J,KP1,1,0)
      DZK =ZL(I,J,K  ,1,0)
      DZM1=ZL(I,J,KM1,1,0)
C ***   SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=XL(I,JP1,K,1,2)
      DXS=XL(I,J  ,K,1,2)
      DXF=XL(I,J,KP1,1,3)
      DXB=XL(I,J,K  ,1,3)
      DYF=YL(I,J,KP1,1,3)
      DYB=YL(I,J,K  ,1,3)
      DYE=YL(IP1,J,K,1,1)
      DYW=YL(I  ,J,K,1,1)
      DZE=ZL(IP1,J,K,1,1)
      DZW=ZL(I  ,J,K,1,1)
      DZN=ZL(I,JP1,K,1,2)
      DZS=ZL(I,J  ,K,1,2)
C ***   CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR U
      DXEE=XL(IP2,J,K,1,1)
      DXE =XL(IP1,J,K,1,1)
      DXW =XL(I  ,J,K,1,1)
      DXWW=XL(IM1,J,K,1,1)
      DYNN=YL(I,JP2,K,1,2)
      DYN =YL(I,JP1,K,1,2)
      DYS =YL(I,J  ,K,1,2)
      DYSS=YL(I,JM1,K,1,2)
```

```
      DZFF=ZL(I,J,KP2,1,3)
      DZF =ZL(I,J,KP1,1,3)
      DZB =ZL(I,J,K  ,1,3)
      DZBB=ZL(I,J,KM1,1,3)
C ***  DEFINE THE AREA OF THE CONTROL VOLUME
      DXYF=DXF*DYF
      DXYB=DXB*DYB
      DYZE=DYE*DZE
      DYZW=DYW*DZW
      DZXN=DZN*DXN
      DZXS=DZS*DXS
      VOL=DXI*DYJ*DZK
      VOLDT=VOL/DTIME
      ZXOYN=DZXN/DYN
      ZXOYS=DZXS/DYS
      XYOZF=DXYF/DZF
      XYOZB=DXYB/DZB
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW

C ***    USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE
C        PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.

      GNE=SILIN(R(I  ,JP1,K),R(I  ,J,K),DYP1,DYJ)*V(I  ,JP1,K)
      GNW=SILIN(R(IM1,JP1,K),R(IM1,J,K),DYP1,DYJ)*V(IM1,JP1,K)
      GSE=SILIN(R(I  ,JM1,K),R(I  ,J,K),DYM1,DYJ)*V(I  ,J  ,K)
      GSW=SILIN(R(IM1,JM1,K),R(IM1,J,K),DYM1,DYJ)*V(IM1,J  ,K)
      GE =SILIN(R(IP1,J,K),R(I  ,J,K),DXEE,DXE)*U(IP1,J,K)
      GP =SILIN(R(IM1,J,K),R(I  ,J,K),DXW ,DXE)*U(I  ,J,K)
      GW =SILIN(R(IM2,J,K),R(IM1,J,K),DXWW,DXW)*U(IM1,J,K)
      GFE=SILIN(R(I  ,J,KP1),R(I  ,J,K),DZP1,DZK)*W(I  ,J,KP1)
      GFW=SILIN(R(IM1,J,KP1),R(IM1,J,K),DZP1,DZK)*W(IM1,J,KP1)
      GBE=SILIN(R(I  ,J,KM1),R(I  ,J,K),DZM1,DZK)*W(I  ,J,K  )
      GBW=SILIN(R(IM1,J,KM1),R(IM1,J,K),DZM1,DZK)*W(IM1,J,K  )
      CE=0.5*(GE+GP)*DYZE
      CW=0.5*(GP+GW)*DYZW
      CN=SILIN(GNE,GNW,DXE,DXW)*DZXN
      CS=SILIN(GSE,GSW,DXE,DXW)*DZXS
      CF=SILIN(GFE,GFW,DXE,DXW)*DXYF
      CB=SILIN(GBE,GBW,DXE,DXW)*DXYB
      VISE=VIS(I  ,J,K)
      VISW=VIS(IM1,J,K)
      VISN=     (VIS(I  ,JP1,K)+VIS(I  ,J,K)+
     &           VIS(IM1,JP1,K)+VIS(IM1,J,K))/4.0
      VISS=     (VIS(I  ,JM1,K)+VIS(I  ,J,K)+
     &           VIS(IM1,JM1,K)+VIS(IM1,J,K))/4.0
      VISF=     (VIS(I  ,J,KP1)+VIS(I  ,J.K)+
     &           VIS(IM1,J,KP1)+VIS(IM1,J,K))/4.0
      VISB=     (VIS(I  ,J,KM1)+VIS(I  ,J,K)+
     &           VIS(IM1,J,KM1)+VIS(IM1,J,K))/4.0

      VISN1=ZXOYN*VISN
      VISS1=ZXOYS*VISS
      VISE1=YZOXE*VISE
      VISW1=YZOXW*VISW
      VISF1=XYOZF*VISF
      VISB1=XYOZB*VISB
      CEP=(ABS(CE)+CE)*DXE/DXI/16.
      CEM=(ABS(CE)-CE)*DXE/DXP1/16.
      CWP=(ABS(CW)+CW)*DXW/DXM1/16.
      CWM=(ABS(CW)-CW)*DXW/DXI/16.
```

```
      CNP=(ABS(CN)+CN)*DYN/DYJ/16.
      CNM=(ABS(CN)-CN)*DYN/DYP1/16.
      CSP=(ABS(CS)+CS)*DYS/DYM1/16.
      CSM=(ABS(CS)-CS)*DYS/DYJ/16.
      CFP=(ABS(CF)+CF)*DZF/DZK/16.
      CFM=(ABS(CF)-CF)*DZF/DZP1/16.
      CBP=(ABS(CB)+CB)*DZB/DZM1/16.
      CBM=(ABS(CB)-CB)*DZB/DZK/16.
      AE(I,J,K)=-.5*CE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE+VISE1
      AW(I,J,K)= .5*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW+VISW1
      AN(I,J,K)=-.5*CN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN+VISN1
      AS(I,J,K)= .5*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS+VISS1
      AF(I,J,K)=-.5*CF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF+VISF1
      AB(I,J,K)= .5*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB+VISB1
  801 AEE=-CEM*DXE/DXEE
      AEER=AEE*UPD(IP2,J,K)
  802 CONTINUE
  803 AWW=-CWP*DXW/DXWW
      AWWR=AWW*UPD(IM2,J,K)
  804 CONTINUE
      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
  805 ANN=-CNM*DYN/DYNN
      ANNR=ANN*UPD(I,JP2,K)
  806 CONTINUE
      IF (J.GT.2) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
  807 ASS=-CSP*DYS/DYSS
      ASSR=ASS*UPD(I,JM2,K)
  808 CONTINUE
      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*UPD(I,J,KP2)
  810 CONTINUE
      IF (K.GT.2) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*UPD(I,J,KM2)
  812 CONTINUE

C ##############################################################
C ##############################################################
C *** MODIFICATION FOR DECK    BOUNDARIES
  900 CONTINUE
      IF (NOD(IM2,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0
  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
      AEE=0.0
      AEER=0.0
  902 CONTINUE
      IF (NOD(I,JM1,K).EQ.0) GOTO 903
      ASS=0.0
```

```
      ASSR=0.0
  903 CONTINUE
      IF (NOD(I,JP1,K).EQ.0) GOTO 904
      ANN=0.0
      ANNR=0.0
  904 CONTINUE
      IF (NOD(I,J,KM1).EQ.0) GOTO 905
      ABB=0.0
      ABBR=0.0
  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0
  906 CONTINUE
C ######################################################################
C ######################################################################


C ***   SU FROM NORMAL STRESS
      RE=(SIG11(I  ,J,K)-(U(IP1,J,K)-U(I  ,J,K))*VISE/DXE)*DYZE
      RW=(SIG11(IM1,J,K)-(U(I  ,J,K)-U(IM1,J,K))*VISW/DXW)*DYZW
      RN=(SIG12(I,JP1,K)-(U(I,JP1,K)-U(I,J  ,K))*VISN/DYN)*DZXN
      RS=(SIG12(I,J  ,K)-(U(I,J  ,K)-U(I,JM1,K))*VISS/DYS)*DZXS
      RF=(SIG13(I,J,KP1)-(U(I,J,KP1)-U(I,J,K  ))*VISF/DZF)*DXYF
      RB=(SIG13(I,J,K  )-(U(I,J,K  )-U(I,J,KM1))*VISB/DZB)*DXYB
C ***       SU FROM CURVED STRESSES AND ACCELERATIONS
      AVG12=0.5*(SIG12(I,JP1,K)+SIG12(I,J,K))
      AVG13=0.5*(SIG13(I,J,KP1)+SIG13(I,J,K))
      AVG22=SILIN(SIG22(I,J,K),SIG22(IM1,J,K),DXE,DXW)
      AVG33=SILIN(SIG33(I,J,K),SIG33(IM1,J,K),DXE,DXW)
      AU1=U(I,J,K)
      AU2=BILIN(V(I  ,JP1,K),V(I  ,J,K),DYJ,DYJ,
     &          V(IM1,JP1,K),V(IM1,J,K),DYJ,DYJ, DXE,DXW)
      AU3=BILIN(W(I  ,J,KP1),W(I  ,J,K),DZK,DZK,
     &          W(IM1,J,KP1),W(IM1,J,K),DZK,DZK, DXE,DXW)
      AR=SILIN(R(I,J,K),R(IM1,J,K),DXE,DXW)
      ARU12=AR*AU1*AU2
      ARU13=AR*AU1*AU3
      ARU22=AR*AU2*AU2
      ARU33=AR*AU3*AU3
      RRY=(AVG12-ARU12)*DZK*(DXN-DXS)
      RRZ=(AVG13-ARU13)*DYJ*(DXF-DXB)
      RRX=(AVG22-ARU22)*DZK*(DYE-DYW)+
     &    (AVG33-ARU33)*DYJ*(DZE-DZW)
      AP(I,J,K)=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)
     &         +AF(I,J,K)+AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB
      SP(I,J,K)=-(ROD(I,J,K)*DXW+ROD(IM1,J,K)*DXE)/(DXW+DXE)*VOLDT
      SU(I,J,K)= (ROD(I,J,K)*DXW+ROD(IM1,J,K)*DXE)/(DXW+DXE)*VOLDT
     &          *UOD(I,J,K)
      SU(I,J,K)=SU(I,J,K)+DYJ*DZK*(P(IM1,J,K)-P(I,J,K))
     &         +AEER+AWWR+ANNR+ASSR+AFFR+ABBR
     &         +RE-RW+RN-RS+RF-RB+RRY+RRZ-RRX
     &-BUOY*SIN(ZC(K))*((R(I,J,K)-REQ(I,J,K))*DXW*COS(XC(I))+(R(IM1,
     & J,K)-REQ(IM1,J,K))*DXE*COS(XC(IM1)))/(DXW+DXE)*VOL
  100 CONTINUE
C ***   TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU
C
C ***   RADIUS DIRECTION
      DO 500 K=2,NK
      DO 500 I=2,NI
CC    SP(I,2,K)=SP(I,2,K)+AS(I,2,K)
      SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
```

```
      SU(I,2,K)=SU(I,2,K)+2.0*U(I,1,K)*AS(I,2,K)
      SP(I,NJ,K)=SP(I,NJ,K)-AN(I,NJ,K)
      AN(I,NJ,K)=0.
      AS(I,2,K)=0.
  500 CONTINUE
C ***    CYLIC CONDITION
      DO 502 K=2,NK
      DO 502 J=2,NJ
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*U(1    ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*U(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
  502 CONTINUE
C ***      FRONT AND BACK WALLS
      DO 600 I=2,NI
      DO 600 J=2,NJ
C ***         SLIP WALLS
      SP(I,J,2)=SP(I,J,2)+AB(I,J,2)
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)
      AF(I,J,NK)=0.
      AB(I,J,2)=0.
 600  CONTINUE



      IF (NCHIP.EQ.0) GOTO 105
C ##############################################################
C ################################################
C *** MODIFICATION FOR DECK BOUNDARIES
      DO 101 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      IBM1=IB-1
      IEP1=IE+1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      JBM1=JB-1
      JEP1=JE+1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      KBM1=KB-1
      KEP1=KE+1
      DO 102 J=JB,JE-1
      DO 102 K=KB,KE-1
      AE(IBM1,J,K)=0.0
      AW(IEP1,J,K)=0.0
  102 CONTINUE
      DO 103 I=IB,IE
      DO 103 K=KB,KE-1
      SP(I,JBM1,K)=SP(I,JBM1,K)-AN(I,JBM1,K)
      AN(I,JBM1,K)=0.0
      SP(I,JE,K)=SP(I,JE,K)-AS(I,JE,K)
      AS(I,JE,K)=0.0
  103 CONTINUE
      DO 106 I=IB,IE
      DO 106 J=JB,JE-1
      SP(I,J,KBM1)=SP(I,J,KBM1)-AF(I,J,KBM1)
      AF(I,J,KBM1)=0.0
      SP(I,J,KE)=SP(I,J,KE)-AB(I,J,KE)
      AB(I,J,KE)=0.0
  106 CONTINUE
```

```
C *** FOR THE CELLS INSIDE OF THE DECKS
       DO 104 I=IB,IE
       DO 104 J=JB,JE-1
       DO 104 K=KB,KE-1
       SP(I,J,K)=-1.0E20
       AW(I,J,K)=0.
       AE(I,J,K)=0.
       AS(I,J,K)=0.
       AN(I,J,K)=0.
       SU(I,J,K)=0.
  104 CONTINUE
  101 CONTINUE
  105 CONTINUE
C #########################################################
C #########################################################


C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
       DO 301 K=2,NK
       DO 301 J=2,NJ
       DO 301 I=2,NI
       DYJ=YL(I,J,K,1,0)
       DZK=ZL(I,J,K,1,0)
       DYZ=DYJ*DZK
       AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
       DU(I,J,K)=DYZ/AP(I,J,K)
  301 CONTINUE


C ***   SOLVE FOR U
       CALL TRID (2,2,2,NI,NJ,NK,U)
       DO 74 I=2,NIP1
       DO 74 J=2,NJP1
       U(I,J,1)=U(I,J,2)
       U(I,J,NKP1)=U(I,J,NK)
  74  CONTINUE

       DO 79 I=1,NIP1
       DO 79 K=1,NKP1
C     U(I,1,K)=U(I,2,K)
  79  CONTINUE

       IF (NCHIP.EQ.0) GOTO 112
C ############################################################
C ##########################################
C *** RESET THE VELOCITY INSIDE OF DECK
       DO 110 N=1,NCHIP
       IB=ICHPB(N)
       IE=IB+NCHPI(N)-1
       JB=JCHPB(N)
       JE=JB+NCHPJ(N)-1
       KB=KCHPB(N)
       KE=KB+NCHPK(N)-1
       DO 108 I=IB,IE
       DO 108 J=JB,JE-1
       DO 108 K=KB,KE-1
       U(I,J,K)=0.0
  108 CONTINUE
  110 CONTINUE
  112 CONTINUE
C ################################################
C ##############################################################
       RETURN
       END
```

```
C
C     ******************************************************************
      SUBROUTINE CALV
C     ******************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &          ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &          NCHPK(10),TCHP(10),CPS(10),CONS(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)

C ***   CALCULATE COEFFICIENTS
      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=3,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2) IM2=NIM1
      IF (I.EQ.NI) IP2=3

C       CENTRAL LENGTH OF THE SCALE CONTROL VOLUME
      DXP1=XL(IP1,J,K,2,0)
      DXI =XL(I  ,J,K,2,0)
      DXM1=XL(IM1,J,K,2,0)
      DYP1=YL(I,JP1,K,2,0)
      DYJ =YL(I,J  ,K,2,0)
      DYM1=YL(I,JM1,K,2,0)
      DZP1=ZL(I,J,KP1,2,0)
      DZK =ZL(I,J,K  ,2,0)
      DZM1=ZL(I,J,KM1,2,0)
C ***   SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=XL(I,JP1,K,2,2)
      DXS=XL(I,J  ,K,2,2)
```

```
      DXF=XL(I,J,KP1,2,3)
      DXB=XL(I,J,K  ,2,3)

      DYF=YL(I,J,KP1,2,3)
      DYB=YL(I,J,K  ,2,3)
      DYE=YL(IP1,J,K,2,1)
      DYW=YL(I  ,J,K,2,1)

      DZE=ZL(IP1,J,K,2,1)
      DZW=ZL(I  ,J,K,2,1)
      DZN=ZL(I,JP1,K,2,2)
      DZS=ZL(I,J  ,K,2,2)
C ***    CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME
      DXEE=XL(IP2,J,K,2,1)
      DXE =XL(IP1,J,K,2,1)
      DXW =XL(I  ,J,K,2,1)
      DXWW=XL(IM1,J,K,2,1)

      DYNN=YL(I,JP2,K,2,2)
      DYN =YL(I,JP1,K,2,2)
      DYS =YL(I,J  ,K,2,2)
      DYSS=YL(I,JM1,K,2,2)

      DZFF=ZL(I,J,KP2,2,3)
      DZF =ZL(I,J,KP1,2,3)
      DZB =ZL(I,J,K  ,2,3)
      DZBB=ZL(I,J,KM1,2,3)
C ***   DEFINE THE AREA OF THE CONTROL VOLUME
      DXYF=DXF*DYF
      DXYB=DXB*DYB
      DYZE=DYE*DZE
      DYZW=DYW*DZW
      DZXN=DZN*DXN
      DZXS=DZS*DXS

      VOL=DXI*DYJ*DZK
      VOLDT=VOL/DTIME

      ZXOYN=DZXN/DYN
      ZXOYS=DZXS/DYS
      XYOZF=DXYF/DZF
      XYOZB=DXYB/DZB
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW


C ***     USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE
C    &    PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.

      GEN=SILIN(R(IP1,J  ,K),R(I,J  ,K),DXP1,DXI)*U(IP1,J  ,K)
      GES=SILIN(R(IP1,JM1,K),R(I,JM1,K),DXP1,DXI)*U(IP1,JM1,K)
      GWN=SILIN(R(IM1,J  ,K),R(I,J  ,K),DXM1,DXI)*U(I  ,J  ,K)
      GWS=SILIN(R(IM1,JM1,K),R(I,JM1,K),DXM1,DXI)*U(I  ,JM1,K)

      GN =SILIN(R(I,JP1,K),R(I,J  ,K),DYNN,DYN)*V(1,JP1,K)
      GP =SILIN(R(I,JM1,K),R(I,J  ,K),DYS ,DYN)*V(I,J  ,K)
      GS =SILIN(R(I,JM2,K),R(I,JM1,K),DYSS,DYS)*V(I,JM1,K)

      GFN=SILIN(R(I,J  ,KP1),R(I,J  ,K),DZP1,DZK)*W(I,J  ,KP1)
      GFS=SILIN(R(I,JM1,KP1),R(I,JM1,K),DZP1,DZK)*W(I,JM1,KP1)
      GBN=SILIN(R(I,J  ,KM1),R(I,J  ,K),DZM1,DZK)*W(I,J  ,K  )
      GBS=SILIN(R(I,JM1,KM1),R(I,JM1,K),DZM1,DZK)*W(I,JM1,K  )

      CN=0.5*(GN+GP)*DZXN
      CS=0.5*(GP+GS)*DZXS

      CE=SILIN(GEN,GES,DYN,DYS)*DYZE
      CW=SILIN(GWN,GWS,DYN,DYS)*DYZW

      CF=SILIN(GFN,GFS,DYN,DYS)*DXYF
      CB=SILIN(GBN,GBS,DYN,DYS)*DXYB

      VISN=VIS(I,J  ,K)
      VISS=VIS(I,JM1,K)
```

```
      VISE=     (VIS(IP1,J  ,K)+VIS(I,J  ,K)+
     &           VIS(IP1,JM1,K)+VIS(I,JM1,K))/4.0
      VISW=     (VIS(IM1,J  ,K)+VIS(I,J  ,K)+
     &           VIS(IM1,JM1,K)+VIS(I,JM1,K))/4.0
      VISF=     (VIS(I,J  ,KP1)+VIS(I,J  ,K)+
     &           VIS(I,JM1,KP1)+VIS(I,JM1,K))/4.0
      VISB=     (VIS(I,J  ,KM1)+VIS(I,J  ,K)+
     &           VIS(I,JM1,KM1)+VIS(I,JM1,K))/4.0

      VISN1=ZXOYN*VISN
      VISS1=ZXOYS*VISS
      VISE1=YZOXE*VISE
      VISW1=YZOXW*VISW
      VISF1=XYOZF*VISF
      VISB1=XYOZB*VISB

      CEP=(ABS(CE)+CE)*DXE/DXI/16.
      CEM=(ABS(CE)-CE)*DXE/DXP1/16.
      CWP=(ABS(CW)+CW)*DXW/DXM1/16.
      CWM=(ABS(CW)-CW)*DXW/DXI/16.
      CNP=(ABS(CN)+CN)*DYN/DYJ/16.
      CNM=(ABS(CN)-CN)*DYN/DYP1/16.
      CSP=(ABS(CS)+CS)*DYS/DYM1/16.
      CSM=(ABS(CS)-CS)*DYS/DYJ/16.
      CFP=(ABS(CF)+CF)*DZF/DZK/16.
      CFM=(ABS(CF)-CF)*DZF/DZP1/16.
      CBP=(ABS(CB)+CB)*DZB/DZM1/16.
      CBM=(ABS(CB)-CB)*DZB/DZK/16.
      AE(I,J,K)=-.5*CE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE+VISE1
      AW(I,J,K)= .5*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW+VISW1
      AN(I,J,K)=-.5*CN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN+VISN1
      AS(I,J,K)= .5*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS+VISS1
      AF(I,J,K)=-.5*CF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF+VISF1
      AB(I,J,K)= .5*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB+VISB1

  801 AEE=-CEM*DXE/DXEE
      AEER=AEE*VPD(IP2,J,K)
  802 CONTINUE
  803 AWW=-CWP*DXW/DXWW
      AWWR=AWW*VPD(IM2,J,K)
  804 CONTINUE
      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
  805 ANN=-CNM*DYN/DYNN
      ANNR=ANN*VPD(I,JP2,K)
  806 CONTINUE
      IF (J.GT.3) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
  807 ASS=-CSP*DYS/DYSS
      ASSR=ASS*VPD(I,JM2,K)
  808 CONTINUE
      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*VPD(I,J,KP2)
  810 CONTINUE
```

```
      IF (K.GT.2) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*VPD(I,J,KM2)
  812 CONTINUE


C ##############################################################
C #####################################################
C *** MODIFICATION FOR DECK    BOUNDARIES
  900 CONTINUE
      IF (NOD(IM1,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0
  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
      AEE=0.0
      AEER=0.0
  902 CONTINUE
      IF (NOD(I,JM2,K).EQ.0) GOTO 903
      ASS=0.0
      ASSR=0.0
  903 CONTINUE
      IF (NOD(I,JP1,K).EQ.0) GOTO 904
      ANN=0.0
      ANNR=0.0
  904 CONTINUE
      IF (NOD(I,J,KM1).EQ.0) GOTO 905
      ABB=0.0
      ABBR=0.0
  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0
  906 CONTINUE
C ##################################################
C ################################################################

C ***    SU FROM NORMAL STRESS
      RN=(SIG22(I,J  ,K)-(V(I,JP1,K)-V(I,J  ,K))*VISN/DYN)*DZXN
      RS=(SIG22(I,JM1,K)-(V(I,J  ,K)-V(I,JM1,K))*VISS/DYS)*DZXS
      RE=(SIG12(IP1,J,K)-(V(IP1,J,K)-V(I,J  ,K))*VISE/DXE)*DYZE
      RW=(SIG12(I  ,J,K)-(V(I  ,J,K)-V(IM1,J,K))*VISW/DXW)*DYZW
      RF=(SIG23(I,J,KP1)-(V(I,J,KP1)-V(I,J,K  ))*VISF/DZF)*DXYF
      RB=(SIG23(I,J,K  )-(V(I,J,K  )-V(I,J,KM1))*VISB/DZB)*DXYB
C ***        SU FROM CURVED STRESSES AND ACCELERATIONS
      AVG12=0.5*(SIG12(IP1,J,K)+SIG12(I,J,K))
      AVG23=0.5*(SIG23(I,J,KP1)+SIG23(I,J,K))
      AVG11=SILIN(SIG11(I,J,K),SIG11(I,JM1,K),DYN,DYS)
      AVG33=SILIN(SIG33(I,J,K),SIG33(I,JM1,K),DYN,DYS)
      AU2=V(I,J,K)
      AU1=BILIN(U(IP1,J  ,K),U(I,J  ,K),DXI,DXI,
     &          U(IP1,JM1,K),U(I,JM1,K),DXI,DXI, DYN,DYS)
      AU3=BILIN(W(I  ,J,KP1),W(I  ,J,K),DZK,DZK,
     &          W(I,JM1,KP1),W(I,JM1,K),DZK,DZK, DYN,DYS)
      AR=SILIN(R(I,J,K),R(I,JM1,K),DYN,DYS)
      ARU12=AR*AU1*AU2
      ARU23=AR*AU2*AU3
      ARU11=AR*AU1*AU1
      ARU33=AR*AU3*AU3
```

```
      RRX=(AVG12-ARU12)*DZK*(DYE-DYW)
      RRZ=(AVG23-ARU23)*DXI*(DYF-DYB)
      RRY=(AVG11-ARU11)*DZK*(DXN-DXS)+
     &    (AVG33-ARU33)*DXI*(DZN-DZS)


      AP(I,J,K)=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)
     &         +AF(I,J,K)+AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB
      SP(I,J,K)=-(ROD(I,J,K)*DYS+ROD(I,JM1,K)*DYN)/(DYS+DYN)*VOLDT
      SU(I,J,K)= (ROD(I,J,K)*DYS+ROD(I,JM1,K)*DYN)/(DYS+DYN)*VOLDT
     &           *VOD(I,J,K)
      SU(I,J,K)=SU(I,J,K)+DZK*DXI*(P(I,JM1,K)-P(I,J,K))
     &          +AEER+AWWR+ANNR+ASSR+AFFR+ABBR
     &          +RE-RW+RN-RS+RF-RB+RRX+RRZ-RRY
     &   -BUOY*((R(I,J,K)-REQ(I,J,K))*DYS+(R(I,JM1,K)
     &      -REQ(I,JM1,K))*DYN)/(DYS+DYN)*VOL*SIN(ZC(K))*SIN(XC(I))
  100 CONTINUE

C ***    TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU
C
C ***    RADIUS DIRECTION
      DO 500 K=2,NK
      DO 500 I=2,NI
CC    SP(I,3,K)=SP(I,3,K)+AS(I,3,K)
      SU(I,3,K)=SU(I,3,K)+AS(I,3,K)*V(I,2,K)
      AS(I,3,K)=0.
      AN(I,NJ,K)=0.
  500 CONTINUE
C ***    CYLIC CONDITIONS
      DO 502 K=2,NK
      DO 502 J=3,NJ
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*V(1   ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*V(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
  502 CONTINUE
C ***    FRONT AND BACK WALL
      DO 600 I=2,NI
      DO 600 J=3,NJ
      JM1=J-1
C ***          SLIP WALLS
      SP(I,J,2)=SP(I,J,2)+AB(I,J,2)
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)
      AF(I,J,NK)=0.
      AB(I,J,2)=0.
  600 CONTINUE


C ##############################################################
C *** MODIFICATION FOR DECK BOUNDARIES
      DO 101 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      IBM1=IB-1
      IEP1=IE+1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      JBM1=JB-1
      JEP1=JE+1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      KBM1=KB-1
      KEP1=KE+1
```

```
      DO 102 J=JB,JE
      DO 102 K=KB,KE-1
      SP(IBM1,J,K)=SP(IBM1,J,K)-AE(IBM1,J,K)
      AE(IBM1,J,K)=0.0
      SP(IE,J,K)=SP(IE,J,K)-AW(IE,J,K)
      AW(IE,J,K)=0.0
  102 CONTINUE
      DO 103 I=IB,IE-1
      DO 103 K=KB,KE-1
      AN(I,JBM1,K)=0.0
      AS(I,JEP1,K)=0.0
  103 CONTINUE
      DO 106 I=IB,IE-1
      DO 106 J=JB,JE
      SP(I,J,KBM1)=SP(I,J,KBM1)-AF(I,J,KBM1)
      AF(I,J,KBM1)=0.0
      SP(I,J,KE)=SP(I,J,KE)-AB(I,J,KE)
      AB(I,J,KE)=0.0
  106 CONTINUE

C ##################################################
C ###########################################################
C *** MODIFICATION FOR THE CELLS INSIDE OF THE DECKS
      DO 104 I=IB,IE-1
      DO 104 J=JB,JE
      DO 104 K=KB,KE-1
      SP(I,J,K)=-1.0E20
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AS(I,J,K)=0.
      AN(I,J,K)=0.
      SU(I,J,K)=0.
  104 CONTINUE
  101 CONTINUE
  105 CONTINUE


C ########################################
C #########################################################
C
C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
      DO 300 K=2,NK
      DO 300 J=3,NJ
      DO 300 I=2,NI
      DXI=XL(I,J,K,2,0)
      DZK=ZL(I,J,K,2,0)
      DZX=DZK*DXI
      AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
      DV(I,J,K)=DZX/AP(I,J,K)
  300 CONTINUE

C ***   SOLVE FOR V

      CALL TRID (2,3,2,NI,NJ,NK,V)

      DO 74 I=2,NIP1
      DO 74 J=2,NJP1
      V(I,J,1)=V(I,J,2)
      V(I,J,NKP1)=V(I,J,NK)
   74 CONTINUE
      DO 79 I=1,NIP1
      DO 79 K=1,NKP1
C     V(I,2,K)=V(I,3,K)
```

```
   79 CONTINUE

      IF (NCHIP.EQ.0) GOTO 112
C ###############################################################
C ##################################################
C *** RESET THE VELOCITY INSIDE OF THE DECKS
      DO 110 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      DO 108 I=IB,IE-1
      DO 108 J=JB,JE
      DO 108 K=KB,KE-1
      V(I,J,K)=0.0
  108 CONTINUE
  110 CONTINUE
  112 CONTINUE
C ###################################################
C ################################################################
      RETURN
      END


C
C     *****************************************************************
      SUBROUTINE CALW
C     *****************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &          ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &          NCHPK(10),TCHP(10),CPS(10),CONS(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)

C ***       CALCULATE COEFFICIENTS
      DO 100 K=3,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
```

```
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.2) IM2=NIM1
      IF (I.EQ.NI) IP2=3

C      CENTRAL LENGTH OF THE SCALE CONTROL VOLUME
      DXP1=XL(IP1,J,K,3,0)
      DXI =XL(I  ,J,K,3,0)
      DXM1=XL(IM1,J,K,3,0)
      DYP1=YL(I,JP1,K,3,0)
      DYJ =YL(I,J  ,K,3,0)
      DYM1=YL(I,JM1,K,3,0)
      DZP1=ZL(I,J,KP1,3,0)
      DZK =ZL(I,J,K  ,3,0)
      DZM1=ZL(I,J,KM1,3,0)
C ***   SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=XL(I,JP1,K,3,2)
      DXS=XL(I,J  ,K,3,2)
      DXF=XL(I,J,KP1,3,3)
      DXB=XL(I,J,K  ,3,3)
      DYF=YL(I,J,KP1,3,3)
      DYB=YL(I,J,K  ,3,3)
      DYE=YL(IP1,J,K,3,1)
      DYW=YL(I  ,J,K,3,1)
      DZE=ZL(IP1,J,K,3,1)
      DZW=ZL(I  ,J,K,3,1)
      DZN=ZL(I,JP1,K,3,2)
      DZS=ZL(I,J  ,K,3,2)
C ***   CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME
      DXEE=XL(IP2,J,K,3,1)
      DXE =XL(IP1,J,K,3,1)
      DXW =XL(I  ,J,K,3,1)
      DXWW=XL(IM1,J,K,3,1)
      DYNN=YL(I,JP2,K,3,2)
      DYN =YL(I,JP1,K,3,2)
      DYS =YL(I,J  ,K,3,2)
      DYSS=YL(I,JM1,K,3,2)
      DZFF=ZL(I,J,KP2,3,3)
      DZF =ZL(I,J,KP1,3,3)
      DZB =ZL(I,J,K  ,3,3)
      DZBB=ZL(I,J,KM1,3,3)
C ***   DEFINE THE AREA OF THE CONTROL VOLUME
      DXYF=DXF*DYF
      DXYB=DXB*DYB
      DYZE=DYE*DZE
      DYZW=DYW*DZW
      DZXN=DZN*DXN
      DZXS=DZS*DXS
      VOL=DXI*DYJ*DZK
      VOLDT=VOL/DTIME
      ZXOYN=DZXN/DYN
      ZXOYS=DZXS/DYS
      XYOZF=DXYF/DZF
      XYOZB=DXYB/DZB
      YZOXE=DYZE/DXE
      YZOXW=DYZW/DXW
```

```
C ***     USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE
C    &    PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.

      GNF=SILIN(R(I,JP1,K  ),R(I,J,K  ),DYP1,DYJ)*V(I,JP1,K  )
      GNB=SILIN(R(I,JP1,KM1),R(I,J,KM1),DYP1,DYJ)*V(I,JP1,KM1)
      GSF=SILIN(R(I,JM1,K  ),R(I,J,K  ),DYM1,DYJ)*V(I,J  ,K  )
      GSB=SILIN(R(I,JM1,KM1),R(I,J,KM1),DYM1,DYJ)*V(I,J  ,KM1)

      GF =SILIN(R(I,J,KP1),R(I,J,K  ),DZFF,DZF)*W(I,J,KP1)
      GP =SILIN(R(I,J,KM1),R(I,J,K  ),DZB ,DZF)*W(I,J,K  )
      GB =SILIN(R(I,J,KM2),R(I,J,KM1),DZBB,DZB)*W(I,J,KM1)

      GEF=SILIN(R(IP1,J,K  ),R(I,J,K  ),DXP1,DXI)*U(IP1,J,K  )
      GEB=SILIN(R(IP1,J,KM1),R(I,J,KM1),DXP1,DXI)*U(IP1,J,KM1)
      GWF=SILIN(R(IM1,J,K  ),R(I,J,K  ),DXM1,DXI)*U(I  ,J,K  )
      GWB=SILIN(R(IM1,J,KM1),R(I,J,KM1),DXM1,DXI)*U(I  ,J,KM1)

      CF=0.5*(GF+GP)*DXYF
      CB=0.5*(GP+GB)*DXYB

      CN=SILIN(GNF,GNB,DZF,DZB)*DZXN
      CS=SILIN(GSF,GSB,DZF,DZB)*DZXS

      CE=SILIN(GEF,GEB,DZF,DZB)*DYZE
      CW=SILIN(GWF,GWB,DZF,DZB)*DYZW

      VISF=VIS(I,J,K  )
      VISB=VIS(I,J,KM1)

      VISN=     (VIS(I,JP1,K  )+VIS(I,J,K  )+
     &           VIS(I,JP1,KM1)+VIS(I,J,KM1))/4.0
      VISS=     (VIS(I,JM1,K  )+VIS(I,J,K  )+
     &           VIS(I,JM1,KM1)+VIS(I,J,KM1))/4.0

      VISE=     (VIS(IP1,J,K  )+VIS(I,J,K  )+
     &           VIS(IP1,J,KM1)+VIS(I,J,KM1))/4.0
      VISW=     (VIS(IM1,J,K  )+VIS(I,J,K  )+
     &           VIS(IM1,J,KM1)+VIS(I,J,KM1))/4.0


      VISN1=ZXOYN*VISN
      VISS1=ZXOYS*VISS
      VISE1=YZOXE*VISE
      VISW1=YZOXW*VISW
      VISF1=XYOZF*VISF
      VISB1=XYOZB*VISB


      CEP=(ABS(CE)+CE)*DXE/DXI/16.
      CEM=(ABS(CE)-CE)*DXE/DXP1/16.
      CWP=(ABS(CW)+CW)*DXW/DXM1/16.
      CWM=(ABS(CW)-CW)*DXW/DXI/16.

      CNP=(ABS(CN)+CN)*DYN/DYJ/16.
      CNM=(ABS(CN)-CN)*DYN/DYP1/16.
      CSP=(ABS(CS)+CS)*DYS/DYM1/16.
      CSM=(ABS(CS)-CS)*DYS/DYJ/16.

      CFP=(ABS(CF)+CF)*DZF/DZK/16.
      CFM=(ABS(CF)-CF)*DZF/DZP1/16.
      CBP=(ABS(CB)+CB)*DZB/DZM1/16.
      CBM=(ABS(CB)-CB)*DZB/DZK/16.

      AE(I,J,K)=-.5*CE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE+VISE1
      AW(I,J,K)= .5*CW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW+VISW1
      AN(I,J,K)=-.5*CN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN+VISN1
      AS(I,J,K)= .5*CS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS+VISS1
      AF(I,J,K)=-.5*CF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF+VISF1
      AB(I,J,K)= .5*CB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB+VISB1

  801 AEE=-CEM*DXE/DXEE
      AEER=AEE*WPD(IP2,J,K)
  802 CONTINUE

  803 AWW=-CWP*DXW/DXWW
      AWWR=AWW*WPD(IM2,J,K)
```

```
  804 CONTINUE
      IF (J.LT.NJ) GOTO 805
      ANN=0.
      ANNR=0.
      GOTO 806
  805 ANN=-CNM*DYN/DYNN
      ANNR=ANN*WPD(I,JP2,K)
  806 CONTINUE
      IF (J.GT.2) GOTO 807
      ASS=0.
      ASSR=0.
      GOTO 808
  807 ASS=-CSP*DYS/DYSS
      ASSR=ASS*WPD(I,JM2,K)
  808 CONTINUE
      IF (K.LT.NK) GOTO 809
      AFF=0.
      AFFR=0.
      GOTO 810
  809 AFF=-CFM*DZF/DZFF
      AFFR=AFF*WPD(I,J,KP2)
  810 CONTINUE
      IF (K.GT.3) GOTO 811
      ABB=0.
      ABBR=0.
      GOTO 812
  811 ABB=-CBP*DZB/DZBB
      ABBR=ABB*WPD(I,J,KM2)
  812 CONTINUE

C ##################################################################
C ####################################################
C *** MODIFICATION FOR DECK    BOUNDARIES
  900 CONTINUE
      IF (NOD(IM1,J,K).EQ.0) GOTO 901
      AWW=0.0
      AWWR=0.0
  901 CONTINUE
      IF (NOD(IP1,J,K).EQ.0) GOTO 902
      AEE=0.0
      AEER=0.0
  902 CONTINUE
      IF (NOD(I,JM1,K).EQ.0) GOTO 903
      ASS=0.0
      ASSR=0.0
  903 CONTINUE
      IF (NOD(I,JP1,K).EQ.0) GOTO 904
      ANN=0.0
      ANNR=0.0
  904 CONTINUE
      IF (NOD(I,J,KM2).EQ.0) GOTO 905
      ABB=0.0
      ABBR=0.0
  905 CONTINUE
      IF (NOD(I,J,KP1).EQ.0) GOTO 906
      AFF=0.0
      AFFR=0.0
  906 CONTINUE
C ####################################################
C ####################################################################

C ***    SU FROM NORMAL STRESS
      RF=(SIG33(I,J,K  )-(W(I,J,KP1)-W(I,J,K  ))*VISF/DZF)*DXYF
```

```
      RB=(SIG33(I,J,KM1)-(W(I,J,K  )-W(I,J,KM1))*VISB/DZB)*DXYB
      RN=(SIG23(I,JP1,K)-(W(I,JP1,K)-W(I,J   ,K))*VISN/DYN)*DZXN
      RS=(SIG23(I,J   ,K)-(W(I,J   ,K)-W(I,JM1,K))*VISS/DYS)*DZXS
      RE=(SIG13(IP1,J,K)-(W(IP1,J,K)-W(I   ,J,K))*VISE/DXE)*DYZE
      RW=(SIG13(I   ,J,K)-(W(I   ,J,K)-W(IM1,J,K))*VISW/DXW)*DYZW
C ***        SU FROM CURVED STRESSES AND ACCELERATIONS
      AVG23=0.5*(SIG23(I,JP1,K)+SIG23(I,J,K))
      AVG13=0.5*(SIG13(IP1,J,K)+SIG13(I,J,K))
      AVG22=SILIN(SIG22(I,J,K),SIG22(I,J,KM1),DZF,DZB)
      AVG11=SILIN(SIG11(I,J,K),SIG11(I,J,KM1),DZF,DZB)
      AU3=W(I,J,K)
      AU2=BILIN(V(I,JP1,K  ),V(I,J,K  ),DYJ,DYJ,
     &          V(I,JP1,KM1),V(I,J,KM1),DYJ,DYJ, DZF,DZB)
      AU1=BILIN(U(IP1,J,K  ),U(I,J,K  ),DXI,DXI,
     &          U(IP1,J,KM1),U(I,J,KM1),DXI,DXI, DZF,DZB)
      AR=SILIN(R(I,J,K),R(I,J,KM1),DZF,DZB)
      ARU23=AR*AU2*AU3
      ARU13=AR*AU1*AU3
      ARU22=AR*AU2*AU2
      ARU11=AR*AU1*AU1
      RRY=(AVG23-ARU23)*DXI*(DZN-DZS)
      RRX=(AVG13-ARU13)*DYJ*(DZE-DZW)
      RRZ=(AVG22-ARU22)*DXI*(DYF-DYB)+
     &   (AVG11-ARU11)*DYJ*(DXF-DXB)

      AP(I,J,K)=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)
     &         +AF(I,J,K)+AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB
      SP(I,J,K)=-(ROD(I,J,K)*DZB+ROD(I,J,KM1)*DZF)/(DZB+DZF)*VOLDT
      SU(I,J,K)= (ROD(I,J,K)*DZB+ROD(I,J,KM1)*DZF)/(DZB+DZF)*VOLDT
     &           *WOD(I,J,K)
      SU(I,J,K)=SU(I,J,K)+DXI*DYJ*(P(I,J,KM1)-P(I,J,K))
     &         +AEER+AWWR+ANNR+ASSR+AFFR+ABBR
     &         +RE-RW+RN-RS+RF-RB+RRY+RRX-RRZ
     & -BUOY*((R(I,J,K)-REQ(I,J,K))*DZB*COS(ZC(K))+(R(I,J,
     &  KM1)-REQ(I,J,KM1))*DZF*COS(ZC(KM1)))/(DZB+DZF)*VOL*SIN(XC(I))
  100 CONTINUE
C ***     TAKE CARE OF B.C. THRU AN,AS,AE,AW,AP AND SU
C
C ***     RADIUS DIRECTION
      DO 500 K=3,NK
      DO 500 I=2,NI
      KM1=K-1
CC    SP(I,2,K)=SP(I,2,K)+AS(I,2,K)
      SP(I,2,K)=SP(I,2,K)-AS(I,2,K)
      SU(I,2,K)=SU(I,2,K)+2.0*W(I,1,K)*AS(I,2,K)
      SP(I,NJ,K)=SP(I,NJ,K)-AN(I,NJ,K)
      AS(I,2,K)=0.
      AN(I,NJ,K)=0.
  500 CONTINUE
C ***   CYLIC CONDITIONS
      DO 502 K=3,NK
      DO 502 J=2,NJ
      SU(2 ,J,K)=SU(2 ,J,K)+AW(2 ,J,K)*W(1   ,J,K)
      SU(NI,J,K)=SU(NI,J,K)+AE(NI,J,K)*W(NIP1,J,K)
      AW(2 ,J,K)=0.0
      AE(NI,J,K)=0.0
  502 CONTINUE
C ***     FRONT AND BACK WALL
      DO 600 I=2,NI
      DO 600 J=2,NJ
      SP(I,J,NK)=SP(I,J,NK)+AF(I,J,NK)
      SP(I,J,3 )=SP(I,J,3 )+AB(I,J,3 )
      AF(I,J,NK)=0.
      AB(I,J,3)=0.
```

```
  600 CONTINUE

      IF (NCHIP.EQ.0) GOTO 105
C ######################################################
C ##########################################
C *** MODIFICATION FOR DECK BOUNDARIES
      DO 101 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      IBM1=IB-1
      IEP1=IE+1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      JBM1=JB-1
      JEP1=JE+1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      KBM1=KB-1
      KEP1=KE+1
      DO 102 J=JB,JE-1
      DO 102 K=KB,KE
      SP(IBM1,J,K)=SP(IBM1,J,K)-AE(IBM1,J,K)
      AE(IBM1,J,K)=0.0

      SP(IE,J,K)=SP(IE,J,K)-AW(IE,J,K)
      AW(IE,J,K)=0.0
  102 CONTINUE
      DO 103 I=IB,IE-1
      DO 103 K=KB,KE
      SP(I,JBM1,K)=SP(I,JBM1,K)-AN(I,JBM1,K)
      AN(I,JBM1,K)=0.0
      SP(I,JE,K)=SP(I,JE,K)-AS(I,JE,K)
      AS(I,JE,K)=0.0
  103 CONTINUE
      DO 106 I=IB,IE-1
      DO 106 J=JB,JE-1
      AF(I,J,KBM1)=0.0
      AB(I,J,KEP1)=0.0
  106 CONTINUE
C *** FOR THE CELLS INSIDE OF THE DECKS
      DO 104 I=IB,IE-1
      DO 104 J=JB,JE-1
      DO 104 K=KB,KE
      SP(I,J,K)=-1.0E20
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AS(I,J,K)=0.
      AN(I,J,K)=0.
      SU(I,J,K)=0.
  104 CONTINUE
  101 CONTINUE
  105 CONTINUE
C #######################################
C #########################################################

C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS

      DO 301 K=3,NK
      DO 301 J=2,NJ
      DO 301 I=2,NI
      DXI=XL(I,J,K,3,0)
      DYJ=YL(I,J,K,3,0)
      DXY=DXI*DYJ
```

```
      AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
      DW(I,J,K)=DXY/AP(I,J,K)
  301 CONTINUE

C ***   SOLVE FOR W
      CALL TRID  (2,2,3,NI,NJ,NK,W)
C
      DO 76 I=1,NI
      DO 76 J=1,NJ
      W(I,J,2)=W(I,J,3)
      W(I,J,NKP1)=W(I,J,NK)
   76 CONTINUE

      IF (NCHIP.EQ.0) GOTO 112
C ##############################################################
C ##############################################################
C *** RESET THE VELOCITY INSIDE OF THE DECKS
      DO 110 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      DO 108 I=IB,IE-1
      DO 108 J=JB,JE-1
      DO 108 K=KB,KE
      W(I,J,K)=0.0
  108 CONTINUE
  110 CONTINUE
  112 CONTINUE
      RETURN
      END

C    -----------------------------------------------------------
C    ***********************************************************
      SUBROUTINE CALP
C    ***********************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &         DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10)
      COMMON/BL31/ TOD(22,16,32),ROD(22,16,32),POD(22,16,32)
     &  ,COD(22,16,32),UOD(22,16,32),VOD(22,16,32),WOD(22,16,32)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL33/ TPD(22,16,32),RPD(22,16,32),PPD(22,16,32)
     &       ,CPD(22,16,32),UPD(22,16,32),VPD(22,16,32),WPD(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
C ***    CALCULATE COEFFICIENTS
      DO 100 K=2,NK
      KP2=K+2
```

```
      KP1=K+1
      KM1=K-1
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
      IF (I.EQ.NI) IP1=2

C       CENTRAL LENGTH OF THE SCALE CONTROL VOLUME
      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)
      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)
      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)
C ***   SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)
      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)
      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)

C ***   DEFINE AREA OF THE CONTROL VOLUME
      DXYF=DXF*DYF
      DXYB=DXB*DYB
      DYZE=DYE*DZE
      DYZW=DYW*DZW
      DZXN=DZN*DXN
      DZXS=DZS*DXS
      VOL=DXI*DYJ*DZK
      VOLDT=VOL/DTIME
      RN=(R(I,J,K)*DYP1+R(I,JP1,K)*DYJ)/(DYP1+DYJ)
      RS=(R(I,J,K)*DYM1+R(I,JM1,K)*DYJ)/(DYM1+DYJ)
      RE=(R(I,J,K)*DXP1+R(IP1,J,K)*DXI)/(DXP1+DXI)
      RW=(R(I,J,K)*DXM1+R(IM1,J,K)*DXI)/(DXM1+DXI)
      RF=(R(I,J,K)*DZP1+R(I,J,KP1)*DZK)/(DZP1+DZK)
      RB=(R(I,J,K)*DZM1+R(I,J,KM1)*DZK)/(DZM1+DZK)
C ***   DU ON VERTICAL WALLS AND DV ON HORIZENTAL WALLS ARE ZERO
      AN(I,J,K)=RN*DZXN*DV(I,JP1,K)
      AS(I,J,K)=RS*DZXS*DV(I,J,K)
      AE(I,J,K)=RE*DYZE*DU(IP1,J,K)
      AW(I,J,K)=RW*DYZW*DU(I,J,K)
      AF(I,J,K)=RF*DXYF*DW(I,J,KP1)
      AB(I,J,K)=RB*DXYB*DW(I,J,K)
      CN=RN*V(I,JP1,K)*DZXN
```

```
      CS=RS*V(I,J  ,K)*DZXS
      CE=RE*U(IP1,J,K)*DYZE
      CW=RW*U(I  ,J,K)*DYZW
      CF=RF*W(I,J,KP1)*DXYF
      CB=RB*W(I,J,K  )*DXYB
      SMP(I,J,K)=-(R(I,J,K)-ROD(I,J,K))*VOL/DTIME-CE+CW-CN+CS-CF+CB
C     SMP(I,J,K)=-CE+CW-CN+CS-CF+CB
      SU(I,J,K)=SMP(I,J,K)
      SP(I,J,K)=0.
  100 CONTINUE
C ***    TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU
C
C ***    RADIUS DIRECTION
      DO 500 K=2,NK
      DO 500 I=2,NI
      AS(I,2,K)=0.
      AN(I,NJ,K)=0.
  500 CONTINUE
C ***    LEFT WALL AND RIGHT WALL
      DO 501 K=2,NK
      DO 501 J=2,NJ
C     AW(2,J,K)=0.
C     AE(NI,J,K)=0.
  501 CONTINUE
C ***     FRONT AND BACK WALL
      DO 502 I=2,NI
      DO 502 J=2,NJ
      AB(I,J,2)=0.0
      AF(I,J,NK)=0.0
 502  CONTINUE



      IF (NCHIP.EQ.0) GOTO 105
C ############################################################
C ############################################
C *** MODIFICATION FOR DECK BOUNDARIES
      DO 101 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      IBM1=IB-1
      IEP1=IE+1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      JBM1=JB-1
      JEP1=JE+1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      KBM1=KB-1
      KEP1=KE+1
      DO 102 J=JB,JE-1
      DO 102 K=KB,KE-1
      AE(IBM1,J,K)=0.0
      AW(IE,J,K)=0.0
  102 CONTINUE
      DO 103 I=IB,IE-1
      DO 103 K=KB,KE-1
      AN(I,JBM1,K)=0.0
      AS(I,JE,K)=0.0
  103 CONTINUE
      DO 106 I=IB,IE-1
      DO 106 J=JB,JE-1
      AF(I,J,KBM1)=0.0
```

```
      AB(I,J,KE)=0.0
  106 CONTINUE
C *** FOR THE CELLS INSIDE OF THE DECKS
      DO 104 I=IB,IE-1
      DO 104 J=JB,JE-1
      DO 104 K=KB,KE-1
      SP(I,J,K)=-1.0E20
      AW(I,J,K)=0.
      AE(I,J,K)=0.
      AS(I,J,K)=0.
      AN(I,J,K)=0.
      SU(I,J,K)=0.
  104 CONTINUE
  101 CONTINUE
  105 CONTINUE

C ###############################################
C ###################################################################


C ***    ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
      DO 300 J=2,NJ
      DO 300 I=2,NI
      DO 300 K=2,NK
      AP(I,J,K)=AN(I,J,K)+AS(I,J,K)+AE(I,J,K)+AW(I,J,K)-SP(I,J,K)
     &    +AF(I,J,K)+AB(I,J,K)
  300 CONTINUE
C ***    SOLUTION OF FINITE DIFFERENCE EQUATION
      CALL TRID  (2,2,2,NI,NJ,NK,PP)
C *** THIS IS FOR CKECKING


      DO 161 I=1,NIP1
C     WRITE (6,*) I
 949  FORMAT ( '  AW ')
C     WRITE (6,949)
C     WRITE (6,999) ((AW(I,J,K),K=1,NKP1),J=1,NJP1)
 161  CONTINUE
      DO 160 I=1,NIP1
C     WRITE (6,*) I
 948  FORMAT ( '  AE ')
C     WRITE (6,948)
C     WRITE (6,999) ((AE(I,J,K),K=1,NKP1),J=1,NJP1)
 160  CONTINUE
      DO 170 I=1,NIP1
C     WRITE (6,*) I
 958  FORMAT ( '  AB ')
C     WRITE (6,958)
C     WRITE (6,999) ((AB(I,J,K),K=1,NKP1),J=1,NJP1)
 170  CONTINUE
      DO 180 I=1,NIP1
C     WRITE (6,*) I
 968  FORMAT ( '  AF ')
C     WRITE (6,968)
C     WRITE (6,999) ((AF(I,J,K),K=1,NKP1),J=1,NJP1)
 180  CONTINUE
C     WRITE (6,999) ((SU(I,5,K),K=1,NKP1),I=1,NIP1)
      DO 190 I=1,NIP1
C     WRITE (6,*) I
 978  FORMAT ( '  SU ')
C     WRITE (6,978)
C     WRITE (6,999) ((SU(I,J,K),K=1,NKP1),J=1,NJP1)
 190  CONTINUE
      DO 191 I=1,NIP1
C     WRITE (6,*) I
C     WRITE (6,988)
```

```
 988  FORMAT ( '  PP ')
C     WRITE (6,999) ((PP(I,J,K),J=1,NJP1),K=7,7)
 191  CONTINUE
 999  FORMAT (12E10.3)


C ***    CORRECT VELOCITIES AND PRESSURE
C
C ***    CORRECTION FOR VELOCITY U
       DO 600 I=2,NI
       IM1=I-1
       IF (I.EQ.2) IM1=NI
       DO 600 J=2,NJ
       DO 600 K=2,NK
       U(I,J,K)=U(I,J,K)+DU(I,J,K)*(PP(IM1,J,K)-PP(I,J,K))
  600 CONTINUE
C ***    CORRECTION FOR VELOCITY V
       DO 603 J=3,NJ
       JM1=J-1
       DO 603 K=2,NK
       DO 603 I=2,NI
       V(I,J,K)=V(I,J,K)+DV(I,J,K)*(PP(I,JM1,K)-PP(I,J,K))
  603 CONTINUE
C ***   CORRECTION OF VELOCITY W
       DO 604 K=3,NK
       KM1=K-1
       DO 604 I=2,NI
       DO 604 J=2,NJ
       W(I,J,K)=W(I,J,K)+DW(I,J,K)*(PP(I,J,KM1)-PP(I,J,K))
 604  CONTINUE

C ***    CORRECTION FOR PRESSURE P
       DO 606 J=2,NJ
       DO 606 I=1,NIP1
       DO 606 K=1,NK
       P(I,J,K)=P(I,J,K)+PP(I,J,K)
       PP(I,J,K)=0.
  606 CONTINUE
C *** THIS IS FOR R=0.0 CASE
       DO 75 I=1,NIP1
       DO 75 K=1,NKP1
C     U(I,1,K)=U(I,2,K)
C     W(I,1,K)=W(I,2,K)
C     V(I,2,K)=V(I,3,K)
  75  CONTINUE

C *** MODIFICATION FOR R=0.0
C
       DO 55 K=2,NK
       VY=0.0
       VX=0.0
       VZ=0.0
       DO 50 I=2,NI
       VY=VY+U(I,2,K)*COS(XS(I))
       VX=VX-U(I,2,K)*SIN(XS(I))
   50 CONTINUE
       DO 51 I=2,NI
       VY=VY+V(I,3,K)*SIN(XC(I))
       VX=VX+V(I,3,K)*COS(XC(I))
       VZ=VZ+W(I,2,K)
   51 CONTINUE

C *** FIND THE VELOCITIES AT R=0.0
```

```
      DO 52 I=1,NIP1
      U(I,1,K)=(-VX*SIN(XS(I))+VY*COS(XS(I)))/NIM1
      V(I,2,K)= (VX*COS(XC(I))+VY*SIN(XC(I)))/NIM1
      W(I,1,K)=VZ/NIM1
   52 CONTINUE
   55 CONTINUE


C ***  THIS IS FOR THE CYLINDER ONLY (CYLIC CONDITION)
      DO 76 J=1,NJP1
      DO 76 K=1,NKP1
      U(1,J,K)=U(NI,J,K)
      U(NIP1,J,K)=U(2,J,K)
      V(1,J,K)=V(NI,J,K)
      V(NIP1,J,K)=V(2,J,K)
      W(1,J,K)=W(NI,J,K)
      W(NIP1,J,K)=W(2,J,K)
  76  CONTINUE
C ***   THIS FOR SPHERE ONLY
      DO 77 I=1,NIP1
      DO 77 J=1,NJP1
      U(I,J,1)=U(I,J,2)
      V(I,J,1)=V(I,J,2)
      W(I,J,2)=W(I,J,3)
      U(I,J,NKP1)=U(I,J,NK)
      V(I,J,NKP1)=V(I,J,NK)
      W(I,J,NKP1)=W(I,J,NK)
  77  CONTINUE

      IF (NCHIP.EQ.0) GOTO 116
C ##########################################################
C ########################################
C *** RESET THE VELOCITY INSIDE OF DECK
      DO 120 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      DO 109 I=IB,IE
      DO 109 J=JB,JE-1
      DO 109 K=KB,KE-1
      U(I,J,K)=0.0
  109 CONTINUE
      DO 118 I=IB,IE-1
      DO 118 J=JB,JE
      DO 118 K=KB,KE-1
      V(I,J,K)=0.0
  118 CONTINUE
      DO 119 I=IB,IE-1
      DO 119 J=JB,JE-1
      DO 119 K=KB,KE
      W(I,J,K)=0.0
  119 CONTINUE
  120 CONTINUE
  116 CONTINUE
C ###########################################
C ###############################################################
C ***    RECALCULATE THE ERROR SOURCE AFTER CORRECTIONS OF U, V, P
      SORSUM=0.
      RESORM(ITER)=0.
      DO 700 J=2,NJ
      JP1=J+1
```

```
      JM1=J-1
      DO 700 I=2,NI
      IP1=I+1
      IM1=I-1
      DO 700 K=2,NK
      KP1=K+1
      KM1=K-1

C      CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)
      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)
      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)

C ***   SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)
      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)
      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)

C ***   DEFINE AREA OF THE CONTROL VOLUME
      DXYF=DXF*DYF
      DXYB=DXB*DYB
      DYZE=DYE*DZE
      DYZW=DYW*DZW
      DZXN=DZN*DXN
      DZXS=DZS*DXS
      VOL=DXI*DYJ*DZK
      VOLDT=VOL/DTIME


      RN=(R(I,J,K)*DYP1+R(I,JP1,K)*DYJ)/(DYP1+DYJ)
      RS=(R(I,J,K)*DYM1+R(I,JM1,K)*DYJ)/(DYM1+DYJ)
      RE=(R(I,J,K)*DXP1+R(IP1,J,K)*DXI)/(DXP1+DXI)
      RW=(R(I,J,K)*DXM1+R(IM1,J,K)*DXI)/(DXM1+DXI)
      RF=(R(I,J,K)*DZP1+R(I,J,KP1)*DZK)/(DZP1+DZK)
      RB=(R(I,J,K)*DZM1+R(I,J,KM1)*DZK)/(DZM1+DZK)
      CN=RN*V(I,JP1,K)*DZXN
      CS=RS*V(I,J  ,K)*DZXS
      CE=RE*U(IP1,J,K)*DYZE
      CW=RW*U(I  ,J,K)*DYZW
      CF=RF*W(I,J,KP1)*DXYF
      CB=RB*W(I,J,K  )*DXYB
C     SMP(I,J,K)=-CE+CW-CN+CS-CF+CB
      SMP(I,J,K)=-(R(I,J,K)-ROD(I,J,K))*VOL/DTIME-CE+CW-CN+CS-CF+CB
C ***   SORSUM IS ACTUAL MASS INCREASE OR DECREASE FROM CONTINUITY
C       EQUATUON , THIS WILL COMPARE TO SOURCE
      SORSUM=SORSUM+SMP(I,J,K)
C ***   RESORM IS SUM OF THE ABSOLUTE VALUE OF SMP(I,J,K)
```

```
      RESORM(ITER)=RESORM(ITER)+ABS(SMP(I,J,K))
  700 CONTINUE
      RETURN
      END


C ***********************************************************************
      SUBROUTINE TRID(IST,JST,KST,ISP,JSP,KSP,PHI)
C ***********************************************************************
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL36/AP(22,16,32),AE(22,16,32),AW(22,16,32),AN(22,16,32),
     &       AS(22,16,32),AF(22,16,32),AB(22,16,32),
     &     SP(22,16,32),SU(22,16,32),RI(22,16,32)
      DIMENSION A(99),B(99),C(99),PHI(22,16,32)
C     GOTO 405
      ISTM1=IST-1
      A(ISTM1)=0.
      C(ISTM1)=0.
      DO 100 J=JST,JSP
      DO 100 K=KST,KSP
      DO 101 I=IST,ISP
      A(I)=AE(I,J,K)
      B(I)=AW(I,J,K)
      C(I)=AN(I,J,K)*PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)
     &    +AF(I,J,K)*PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
      TERM=1./(AP(I,J,K)-B(I)*A(I-1))
      A(I)=A(I)*TERM
      C(I)=(C(I)+B(I)*C(I-1))*TERM
      IF (ABS(A(I)).LE.1.0E-70) A(I)=0.0
      IF (ABS(B(I)).LE.1.0E-70) B(I)=0.0
      IF (ABS(C(I)).LE.1.0E-70) C(I)=0.0
  101 CONTINUE
      PHI(ISP,J,K)=C(ISP)
      ISTA=IST+1
      DO 102 II=ISTA,ISP
      I=IST+ISP-II
      IP1=I+1
      PHI(I,J,K)=A(I)*PHI(IP1,J,K)+C(I)
  102 CONTINUE
  100 CONTINUE

      DO 2000 J=JST,JSP
      DO 2000 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
 2000 CONTINUE


      JSTM1=JST-1
      A(JSTM1)=0.
      C(JSTM1)=0.
      DO 200 K=KST,KSP
      DO 200 I=IST,ISP
      DO 201 J=JST,JSP
      A(J)=AN(I,J,K)
      B(J)=AS(I,J,K)
      C(J)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)
     &    +AF(I,J,K)*PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
      TERM=1./(AP(I,J,K)-B(J)*A(J-1))
      A(J)=A(J)*TERM
      C(J)=(C(J)+B(J)*C(J-1))*TERM
      IF (ABS(A(J)).LE.1.0E-70) A(J)=0.0
      IF (ABS(B(J)).LE.1.0E-70) B(J)=0.0
      IF (ABS(C(J)).LE.1.0E-70) C(J)=0.0
  201 CONTINUE
      PHI(I,JSP,K)=C(JSP)
      JSTA=JST+1
      DO 202 JJ=JSTA,JSP
```

```
      J=JST+JSP-JJ
      JP1=J+1
      PHI(I,J,K)=A(J)*PHI(I,JP1,K)+C(J)
 202 CONTINUE
 200 CONTINUE
      DO 2001 J=JST,JSP
      DO 2001 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
2001 CONTINUE

      KSTM1=KST-1
      A(KSTM1)=0.
      C(KSTM1)=0.
      DO 300 I=IST,ISP
      DO 300 J=JST,JSP
      DO 301 K=KST,KSP
      A(K)=AF(I,J,K)
      B(K)=AB(I,J,K)
      C(K)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)
     &    +AN(I,J,K)*PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)+SU(I,J,K)
      TERM=1./(AP(I,J,K)-B(K)*A(K-1))
      A(K)=A(K)*TERM
      C(K)=(C(K)+B(K)*C(K-1))*TERM
      IF (ABS(A(K)).LE.1.0E-70) A(K)=0.0
      IF (ABS(B(K)).LE.1.0E-70) B(K)=0.0
      IF (ABS(C(K)).LE.1.0E-70) C(K)=0.0
 301 CONTINUE
      PHI(I,J,KSP)=C(KSP)
      KSTA=KST+1
      DO 302 KK=KSTA,KSP
      K=KST+KSP-KK
      KP1=K+1
      PHI(I,J,K)=A(K)*PHI(I,J,KP1)+C(K)
 302 CONTINUE
 300 CONTINUE
      DO 2002 J=JST,JSP
      DO 2002 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
2002 CONTINUE

      GOTO 700

4405 CONTINUE
 405 KSP1=KSP+1
      B(KSP1)=0.
      C(KSP1)=0.
      DO 600 II=IST,ISP
      I=IST+ISP-II
      DO 600 JJ=JST,JSP
      J=JST+JSP-JJ
      DO 601 KK=KST,KSP
      K=KSP+KST-KK
      KP1=K+1
      A(K)=AF(I,J,K)
      B(K)=AB(I,J,K)
      C(K)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)+AN(I,J,K)*
     &     PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)+SU(I,J,K)
      TERM=1./(AP(I,J,K)-A(K)*B(K+1))
      B(K)=B(K)*TERM
      C(K)=(C(K)+A(K)*C(K+1))*TERM
      IF (ABS(A(K)).LE.1.0E-70) A(K)=0.0
      IF (ABS(B(K)).LE.1.0E-70) B(K)=0.0
      IF (ABS(C(K)).LE.1.0E-70) C(K)=0.0
601  CONTINUE
      PHI(I,J,KST)=C(KST)
      KSTP1=KST+1
```

```
      DO 602 K=KSTP1,KSP
      PHI(I,J,K)=B(K)*PHI(I,J,K-1)+C(K)
 602 CONTINUE
 600 CONTINUE

      DO 2003 J=JST,JSP
      DO 2003 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
2003 CONTINUE


      JSP1=JSP+1
      B(JSP1)=0.
      C(JSP1)=0.
      DO 500 KK=KST,KSP
      K=KST+KSP-KK
      DO 500 II=IST,ISP
      I=IST+ISP-II
      DO 501 JJ=JST,JSP
      J=JSP+JST-JJ
      JP1=J+1
      A(J)=AN(I,J,K)
      B(J)=AS(I,J,K)
      C(J)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)+AF(I,J,K)*
     &     PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
      TERM=1./(AP(I,J,K)-A(J)*B(J+1))
      B(J)=B(J)*TERM
      C(J)=(C(J)+A(J)*C(J+1))*TERM
      IF (ABS(A(J)).LE.1.0E-70) A(J)=0.0
      IF (ABS(B(J)).LE.1.0E-70) B(J)=0.0
      IF (ABS(C(J)).LE.1.0E-70) C(J)=0.0
501  CONTINUE
      PHI(I,JST,K)=C(JST)
      JSTP1=JST+1
      DO 502 J=JSTP1,JSP
      PHI(I,J,K)=B(J)*PHI(I,J-1,K)+C(J)
 502 CONTINUE
 500 CONTINUE

      DO 2004 J=JST,JSP
      DO 2004 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
2004 CONTINUE


      ISP1=ISP+1
      B(ISP1)=0.
      C(ISP1)=0.
      DO 400 JJ=JST,JSP
      J=JST+JSP-JJ
      DO 400 KK=KST,KSP
      K=KST+KSP-KK
      DO 401 II=IST,ISP
      I=ISP+IST-II
      IP1=I+1
      A(I)=AE(I,J,K)
      B(I)=AW(I,J,K)
      C(I)=AN(I,J,K)*PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)+AF(I,J,K)*
     &     PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
      TERM=1./(AP(I,J,K)-A(I)*B(I+1))
      B(I)=B(I)*TERM
      C(I)=(C(I)+A(I)*C(I+1))*TERM
      IF (ABS(A(I)).LE.1.0E-70) A(I)=0.0
      IF (ABS(B(I)).LE.1.0E-70) B(I)=0.0
      IF (ABS(C(I)).LE.1.0E-70) C(I)=0.0
401  CONTINUE
      PHI(IST,J,K)=C(IST)
      ISTP1=IST+1
      DO 402 I=ISTP1,ISP
      PHI(I,J,K)=B(I)*PHI(I-1,J,K)+C(I)
```

```
  402 CONTINUE
  400 CONTINUE
      DO 2005 J=JST,JSP
      DO 2005 K=KST,KSP
      PHI(IST-1,J,K)=PHI(ISP,J,K)
      PHI(ISP+1,J,K)=PHI(IST,J,K)
 2005 CONTINUE

  700 CONTINUE
      RETURN
      END
C     ************************************************************
      BLOCK DATA
C     ************************************************************
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      DATA NIP2,NIP1,NI,NIM1/23,22,21,20/
      DATA NJP2,NJP1,NJ,NJM1/17,16,15,14/
      DATA NKP2,NKP1,NK,NKM1/33,32,31,30/
      DATA NAP1,NA,NAM1,NBP1,NB,NBM1/9,8,7,27,26,25/
      DATA U0,TA,PRT,RHO0,CP0,VIS0,NTMAX0/
     &     1.0,555.86,1.0,0.0714,0.24,1.56E-4,0/
      DATA HCOEF,TINF,CNT,ABTURB,BTURB/12.0,1.0,0.2,2.0,1.0/
      DATA GC,RAIR/32.17,53.34/
      DATA QCORRT,PM1/1.0,0.9/
      END


C   ******************************************************************
      SUBROUTINE GRID
C   ******************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
C ***  RENERATION OF GRID
      PI=4.*ATAN(1.)
      DX=1.0/FLOAT(NIM1)
C     DY=1./FLOAT(NJM1-2)
      DY=1./FLOAT(NJM1-1)
      DZ=PI/FLOAT(NKM1-NB+NA-2)

      DO 19 I=1,NIP2
      XS(I)=(I-2)*DX*2.0*PI
   19 CONTINUE
C     XS(1)=-DX*2.0*PI
C     XS(2)=0.0
C     XS(3)=0.01*2.0*PI
C     DO 19 I=4,13
C     XS(I)=(I-3)*DX*2.0*PI
C 19  CONTINUE
C
C     XS(14)=XS(13)
C     XS(13)=XS(14)-0.01*2.0*PI
C     DO 18 I=15,NIP1
C     XS(I)=XS(14)+(I-14)*DX*2.0*PI
C  18 CONTINUE
C     XS(NIP2)=XS(NIP1)+XS(3)

      YS(1)=0.000
```

```
      YS(2)=0.025
C     YS(3)=0.05
      DO 3 J=3,NJ
      YS(J)=(J-2)*DY
    3 CONTINUE
      YS(NJP1)=YS(NJ)
      YS(NJ  )=YS(NJP1)-3./8./12./9.6
      YS(NJP2)=YS(NJP1)+3./8./12./9.6
CC    DO 3 J=4,NJP2
CC    YS(J)=(J-3)*DY
CC 3  CONTINUE
      DO 4 I=1,NIP1
      IP1=I+1
      DXXC(I)=XS(IP1)-XS(I)
    4 CONTINUE
      DXXC(NIP2)=DXXC(NIP1)
      DO 5 I=2,NIP2
      IM1=I-1
      DXXS(I)=.5*(DXXC(I)+DXXC(IM1))
    5 CONTINUE
      DXXS(1)=DXXS(2)
      DO 7 J=1,NJP1
      JP1=J+1
      DYYC(J)=YS(JP1)-YS(J)
    7 CONTINUE
      DYYC(NJP2)=DYYC(NJP1)
      DO 8 J=2,NJP2
      JM1=J-1
      DYYS(J)=.5*(DYYC(J)+DYYC(JM1))
    8 CONTINUE
      DYYS(1)=DYYS(2)
      DO 20 I=1,NIP2
      XC(I)=XS(I)+DXXC(I)/2.0
   20 CONTINUE
      DO 21 J=1,NJP2
      YC(J)=YS(J)+DYYC(J)/2.0
   21 CONTINUE


      DO 9 K=4,NA
      ZS(K)=(K-3)*DZ
    9 CONTINUE
      DO 30 K=NBP1,NK
      ZS(K)=ZS(NA)+(K-NB)*DZ
   30 CONTINUE
      DO 31 K=NAP1,NB
      ZS(K)=PI/2.
   31 CONTINUE
      ZS(1)=0.0
      ZS(2)=0.05
      ZS(3)=0.10
C     ZS(NKP1)=ZS(NKM1)
C     ZS(NK)=ZS(NKP1)-0.05
C     ZS(NKM1)=ZS(NKP1)-0.10
C     ZS(NKP2)=ZS(NKP1)+0.05
      ZS(NKP2)=ZS(NK)
      ZS(NKP1)=ZS(NKP2)-0.05
      ZS(NK)=ZS(NKP2)-0.10


      DO 10 K=1,NKP1
      IF (K.GE.NA.AND.K.LT.NB) GOTO 10
      KP1=K+1
      DZZC(K)=ZS(KP1)-ZS(K)
   10 CONTINUE
```

```
      DO 32 K=NA,NBM1
      DZZC(K)=2.854/(NB-NA)
   32 CONTINUE
      DZZC(NKP2)=DZZC(NKP1)
      DO 11 K=2,NKP2
      IF (K.EQ.NA.AND.K.EQ.NB) GOTO 11
      KM1=K-1
      DZZS(K)=.5*(DZZC(K)+DZZC(KM1))
  11  CONTINUE
      DZZS(1)=DZZS(2)
      DO 22 K=1,NKP2
      IF (K.GE.NA.AND.K.LT.NB) GOTO 22
      ZC(K)=ZS(K)+DZZC(K)/2.0
  22  CONTINUE
      DO 33 K=NA,NBM1
      ZC(K)=PI/2.
   33 CONTINUE
      IF (YS(1).LT.0.0) YS(1)=0.0
      IF (YC(1).LT.0.0) YC(1)=0.0
      PRINT *
      PRINT *,'     INPUT COORDINATE OF THE TANK IN THE ORDER OF '
      PRINT *,'    I        XS        YS        ZS        XC        YC',
     &    '      ZC        DXXS       DYYS       DZZS       DXXC       '
     & ,  'DYYC       DZZC'
      DO 12 I=1,NKP2
      WRITE(6,102) I,XS(I),YS(I),ZS(I),XC(I),YC(I),ZC(I),
     &              DXXS(I),DYYS(I),DZZS(I),DXXC(I),DYYC(I),DZZC(I)
 102 FORMAT(2X,I4,12(2X,F8.5))
  12  CONTINUE
      RETURN
      END


C    ***********************
      FUNCTION XL(I,J,K,M,N)
C    ***********************
C*************************************************************
C    WHEN M OR N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                    HALF CELL (STAGGERED CELL)                *
C    WHEN M OR N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                    HALF CELL (STAGGERED CELL)                *
C    WHEN M OR N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                    HALF CELL (STAGGERED CELL)                *
C    WHEN M =  N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                    WHOLE CELL                                *
C    WHEN M =  N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                    WHOLE CELL                                *
C    WHEN M =  N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                    WHOLE CELL                                *
C*************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      X1=XC(I)
      X2=YC(J)
      X3=ZC(K)
      DXL=DXXC(I)
      IF(M.EQ.N) GOTO 100
      IF(M.EQ.1.OR.N.EQ.1) X1=XS(I)
      IF(M.EQ.1.OR.N.EQ.1) DXL=DXXS(I)
      IF(M.EQ.2.OR.N.EQ.2) X2=YS(J)
      IF(M.EQ.3.OR.N.EQ.3) X3=ZS(K)
      GOTO 1000
 100 IF(M.EQ.1) X1=XC(I-1)
      IF(M.EQ.1) DXL=DXXC(I-1)
      IF(M.EQ.2) X2=YC(J-1)
```

```
      IF(M.EQ.3) X3=ZC(K-1)
 1000 CONTINUE
      XL=X2*SIN(X3)*DXL
      RETURN
      END

C     ***********************
      FUNCTION YL(I,J,K,M,N)
C     ***********************
C*****************************************************************
C     WHEN M OR N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M OR N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M OR N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M =  N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                     WHOLE CELL                                *
C     WHEN M =  N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                     WHOLE CELL                                *
C     WHEN M =  N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                     WHOLE CELL                                *
C*****************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      X1=XC(I)
      X2=YC(J)
      X3=ZC(K)
      DYL=DYYC(J)
      IF(M.EQ.N) GOTO 100
      IF(M.EQ.2.OR.N.EQ.2) X2=YS(J)
      IF(M.EQ.2.OR.N.EQ.2) DYL=DYYS(J)
      IF(M.EQ.1.OR.N.EQ.1) X1=XS(I)
      IF(M.EQ.3.OR.N.EQ.3) X3=ZS(K)
      GOTO 1000
  100 IF(M.EQ.2) X2=YC(J-1)
      IF(M.EQ.2) DYL=DYYC(J-1)
      IF(M.EQ.1) X1=XC(I-1)
      IF(M.EQ.3) X3=ZC(K-1)
 1000 CONTINUE
      YL=1.00*DYL
      RETURN
      END

C     ***********************
      FUNCTION ZL(I,J,K,M,N)
C     ***********************
C*****************************************************************
C     WHEN M OR N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M OR N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M OR N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                     HALF CELL (STAGGERED CELL)                *
C     WHEN M =  N = 1 THEN SHIFT CELL IN THE NEG X DIRECTION ONE*
C                     WHOLE CELL                                *
C     WHEN M =  N = 2 THEN SHIFT CELL IN THE NEG Y DIRECTION ONE*
C                     WHOLE CELL                                *
C     WHEN M =  N = 3 THEN SHIFT CELL IN THE NEG Z DIRECTION ONE*
C                     WHOLE CELL                                *
C*****************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     & ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      X1=XC(I)
      X2=YC(J)
      X3=ZC(K)
```

```
      DZL=DZZC(K)
      IF(M.EQ.N) GOTO 100
      IF(M.EQ.2.OR.N.EQ.2) X2=YS(J)
      IF(M.EQ.1.OR.N.EQ.1) X1=XS(I)
      IF(M.EQ.3.OR.N.EQ.3) GOTO 200
      GOTO 1000
  200 CONTINUE
      IF (M.EQ.NA.OR.M.EQ.NB) GOTO 2000
      X3=ZS(K)
      DZL=DZZS(K)
      GOTO 1000
  100 IF(M.EQ.3) X3=ZC(K-1)
      IF(M.EQ.3) DZL=DZZC(K-1)
      IF(M.EQ.2) X2=YC(J-1)
      IF(M.EQ.1) X1=XC(I-1)
 1000 CONTINUE
      ZL=X2*DZL
      GOTO 300
 2000 CONTINUE
      DZL1=DZZC(K-1)
      DZL2=DZZC(K)
      IF (M.EQ.NB) DZL1=DZZC(K)
      IF (M.EQ.NB) DZL2=DZZC(K-1)
      ZL=(X2*DZL1+DZL2)/2.
  300 CONTINUE
      RETURN
      END

C     ***************************
      FUNCTION SILIN(V1,V2,D1,D2)
C     ***************************
C     IF (D1.EQ.0.0.AND.D2.EQ.0.0) D1=0.1
C     IF (D1.EQ.0.0.AND.D2.EQ.0.0) D2=0.1
      SILIN=(V1*D2+V2*D1)/(D1+D2)
      RETURN
      END

C     ***********************************************
      FUNCTION BILIN(V1,V2,D1,D2,V3,V4,D3,D4,D5,D6)
C     ***********************************************
      V12=(V1*D2+V2*D1)/(D1+D2)
      V34=(V3*D4+V4*D3)/(D3+D4)
      BILIN=(V12*D6+V34*D5)/(D5+D6)
      END

C     ******************
      SUBROUTINE STRESS
C     ******************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL20/SIG11(22,16,32),SIG12(22,16,32),SIG22(22,16,32)
     &           ,SIG13(22,16,32),SIG23(22,16,32),SIG33(22,16,32)
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &            NCHPK(10),TCHP(10),CPS(10),CONS(10)
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)

      DO 100 K=2,NK
      KP2=K+2
      KP1=K+1
      KM1=K-1
```

```
      KM2=K-2
      DO 100 J=2,NJ
      JP2=J+2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      DO 100 I=2,NI
      IP2=I+2
      IP1=I+1
      IM1=I-1
      IM2=I-2
C      CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
      DXP1=XL(IP1,J,K,0,0)
      DXI =XL(I  ,J,K,0,0)
      DXM1=XL(IM1,J,K,0,0)
      DYP1=YL(I,JP1,K,0,0)
      DYJ =YL(I,J  ,K,0,0)
      DYM1=YL(I,JM1,K,0,0)
      DZP1=ZL(I,J,KP1,0,0)
      DZK =ZL(I,J,K  ,0,0)
      DZM1=ZL(I,J,KM1,0,0)
C ***   SURFACE LENGTH OF THE CONTROL VOLUME
      DXN=XL(I,JP1,K,0,2)
      DXS=XL(I,J  ,K,0,2)
      DXF=XL(I,J,KP1,0,3)
      DXB=XL(I,J,K  ,0,3)
      DYF=YL(I,J,KP1,0,3)
      DYB=YL(I,J,K  ,0,3)
      DYE=YL(IP1,J,K,0,1)
      DYW=YL(I  ,J,K,0,1)
      DZE=ZL(IP1,J,K,0,1)
      DZW=ZL(I  ,J,K,0,1)
      DZN=ZL(I,JP1,K,0,2)
      DZS=ZL(I,J  ,K,0,2)
C ***   CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T
      DXEE=XL(IP2,J,K,0,1)
      DXE =XL(IP1,J,K,0,1)
      DXW =XL(I  ,J,K,0,1)
      DXWW=XL(IM1,J,K,0,1)
      DYNN=YL(I,JP2,K,0,2)
      DYN =YL(I,JP1,K,0,2)
      DYS =YL(I,J  ,K,0,2)
      DYSS=YL(I,JM1,K,0,2)
      DZFF=ZL(I,J,KP2,0,3)
      DZF =ZL(I,J,KP1,0,3)
      DZB =ZL(I,J,K  ,0,3)
      DZBB=ZL(I,J,KM1,0,3)
      UBAR=0.5*(U(IP1,J,K)+U(I,J,K))
      VBAR=0.5*(V(I,JP1,K)+V(I,J,K))
      WBAR=0.5*(W(I,J,KP1)+W(I,J,K))
      DXY=DXI*DYJ
      DYZ=DYJ*DZK
      DZX=DZK*DXI
      SIG11(I,J,K)=2.*VIS(I,J,K)*((U(IP1,J,K)-U(I,J,K))/DXI
     &              +VBAR*(DXN-DXS)/DXY
     &              +WBAR*(DXF-DXB)/DZX)
      SIG22(I,J,K)=2.*VIS(I,J,K)*((V(I,JP1,K)-V(I,J,K))/DYJ
     &              +WBAR*(DYF-DYB)/DYZ
     &              +UBAR*(DYE-DYW)/DXY)
      SIG33(I,J,K)=2.*VIS(I,J,K)*((W(I,J,KP1)-W(I,J,K))/DZK
     &              +UBAR*(DZE-DZW)/DZX
```

```
     &              +VBAR*(DZN-DZS)/DYZ)
 100  CONTINUE
       DO 200 K=2,NKP1
       KP2=K+2
       KP1=K+1
       KM1=K-1
       KM2=K-2
       DO 200 J=2,NJP1
       JP2=J+2
       JP1=J+1
       JM1=J-1
       JM2=J-2
       DO 200 I=2,NIP1
       IP2=I+2
       IP1=I+1
       IM1=I-1
       IM2=I-2

C ****    FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CONTROL
C         VOLUME FOR SIG12
C     IF (J.EQ.2) GOTO 300
       DXN=XL(I,J  ,K,1,0)
       DXS=XL(I,JM1,K,1,0)
       DYE=YL(I  ,J,K,2,0)
       DYW=YL(IM1,J,K,2,0)
       DXI=XL(I  ,J,K,1,2)
       DYJ=YL(I  ,J,K,2,1)
       DYN=YL(I,J  ,K,1,0)
       DYS=YL(I,JM1,K,1,0)
       DXE=XL(I  ,J,K,2,0)
       DXW=XL(IM1,J,K,2,0)
       UBAR=SILIN(U(I,J,K),U(I,JM1,K),DYN,DYS)
       VBAR=SILIN(V(I,J,K),V(IM1,J,K),DXE,DXW)
       VIS12=BILIN(VIS(I  ,J,K),VIS(I  ,JM1,K),DYN,DYS,
     &              VIS(IM1,J,K),VIS(IM1,JM1,K),DYN,DYS, DXE,DXW)
       SIG12(I,J,K)=                 VIS12*((V(I,J,K)-V(IM1,J,K))/DXI
     &                               -VBAR*(DYE-DYW)/(DXI*DYJ))
       SIG12(I,J,K)=SIG12(I,J,K)+VIS12*((U(I,J,K)-U(I,JM1,K))/DYJ
     &                               -UBAR*(DXN-DXS)/(DXI*DYJ))
  300 CONTINUE
C ****    FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CONTROL
C         VOLUME FOR SIG13
       DXF=XL(I,J,K  ,1,0)
       DXB=XL(I,J,KM1,1,0)
       DZE=ZL(I  ,J,K,3,0)
       DZW=ZL(IM1,J,K,3,0)
       DXI=XL(I  ,J,K,1,3)
       DZK=ZL(I  ,J,K,3,1)
       DZF=ZL(I,J,K  ,1,0)
       DZB=ZL(I,J,KM1,1,0)
       DXE=XL(I  ,J,K,3,0)
       DXW=XL(IM1,J,K,3,0)
       IF (DZF.EQ.0.0.OR.DZB.EQ.0.0.OR.DZE.EQ.0.0.OR.DZW.EQ.0.0)
     &   WRITE (6,*) I,J,K, DZF,DZB,DZE,DZW
       UBAR=SILIN(U(I,J,K),U(I,J,KM1),DZF,DZB)
       WBAR=SILIN(W(I,J,K),W(IM1,J,K),DXE,DXW)
       VIS13=BILIN(VIS(I  ,J,K),VIS(I  ,J,KM1),DZF,DZB,
     &              VIS(IM1,J,K),VIS(IM1,J,KM1),DZF,DZB, DXE,DXW)
       SIG13(I,J,K)=                 VIS13*((W(I,J,K)-W(IM1,J,K))/DXI
     &                               -WBAR*(DZE-DZW)/(DXI*DZK))
       SIG13(I,J,K)=SIG13(I,J,K)+VIS13*((U(I,J,K)-U(I,J,KM1))/DZK
     &                               -UBAR*(DXF-DXB)/(DXI*DZK))
```

```
C ****    FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CONTROL
C         VOLUME FOR SIG23
       DZN=ZL(I,J  ,K,3,0)
       DZS=ZL(I,JM1,K,3,0)
       DYF=YL(I,J,K  ,2,0)
       DYB=YL(I,J,KM1,2,0)
       DZK=ZL(I,J,K,3,2)
       DYJ=YL(I,J,K,2,3)
       DYN=YL(I,J  ,K,3,0)
       DYS=YL(I,JM1,K,3,0)
       DZF=ZL(I,J,K  ,2,0)
       DZB=ZL(I,J,KM1,2,0)
       WBAR=SILIN(W(I,J,K),W(I,JM1,K),DYN,DYS)
       VBAR=SILIN(V(I,J,K),V(I,J,KM1),DZF,DZB)
       VIS23=BILIN(VIS(I  ,J,K),VIS(I,JM1,K  ),DYN,DYS,
     &            VIS(I,J,KM1),VIS(I,JM1,KM1),DYN,DYS, DZF,DZB)
       SIG23(I,J,K)=                VIS23*((V(I,J,K)-V(I,J,KM1))/DZK
     &                              -VBAR*(DYF-DYB)/(DZK*DYJ))
       SIG23(I,J,K)=SIG23(I,J,K)+VIS23*((W(I,J,K)-W(I,JM1,K))/DYJ
     &                              -WBAR*(DZN-DZS)/(DZK*DYJ))
 200  CONTINUE
       DO 110 I=1,NIP1
       DO 110 J=1,NJP1
C     WRITE (6,998) I,J,SIG11(I,J,5),SIG12(I,J,5),SIG13(I,J,5),
C    &                 SIG22(I,J,5),SIG23(I,J,5),SIG33(I,J,5)
 998  FORMAT (2X,I4,1X,I4,6(1X,E11.4))
 110  CONTINUE
       RETURN
       END


C
***   ********************************************************************
       SUBROUTINE CALQ(LL)
***   ********************************************************************
       COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
       COMMON/BL12/ NWRITE,NTAPE,NTMAXO,NTREAL,TIME,SORSUM,ITER
       COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
       COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
       COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
       COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
C *** IN MANY OF THE FOLLOWING LINES A TEMPORARY CORRECTION FOR
C  *      ADJUSTING QQ TO AGREE WITH THE PRESSURE HAS BEEN APPLIED.
       XTIME=TIME*H/U0
       IF ( LL .EQ. 1) THEN
           IF (XTIME.LT.23.1) THEN
             PCURVE=9.789522E-5*XTIME**2-2.388310E-6*XTIME**3+
     &         REQ(10,9,16)
             DPDT  =9.789522E-5*XTIME*2-2.388310E-6*XTIME**2*3
           ELSE
       PCURVE=0.0052+.81264E-3*XTIME-.22604E-5*XTIME**2+.27262E-8*XTIME**
     &       3-.115621E-11*XTIME**4+REQ(10,9,16)
       DPDT=.81264E-3-.22604E-5*XTIME*2+.27262E-8*XTIME**
     &       2*3.0-.115621E-11*XTIME**3*4
           ENDIF
       QQ=1.0E8*DPDT
       Q=QQ*3.4134/60./60.
       Q=Q*QCORRT
       endif
       if (LL .eq. 2 ) then
C    THIS USES A CURVE FIT THROUGH THE BURNRATE DATA GIVEN BY NRL
```

```
         ITEST = 0
         BURNR1= 5.4576748 +0.18815346*XTIME-.20153996E-03*XTIME**2
         BURNR2= -1.3116787 + .33158595*XTIME-.7342952E-03*XTIME**2
     &        +.50945510E-06*XTIME**3
         IF (XTIME .LT. 100) THEN
           BURNR= BURNR2 + 1.3117-.013117*XTIME
         ELSE
           BURNR = BURNR2
         ENDIF
         IF(XTIME .LE. 300) GO TO 60
         IF(BURNR2 .LT. BURNR1) THEN
             BURNR = (BURNR1 + BURNR2) / 2
             GO TO 60
         ELSE
             IF ( XTIME .LT. 600.0) GO TO 60
             IF (ITEST .EQ. 0) THEN
                 BURNR3 = BURNR2
                 ITEST = 1
             ENDIF
             BURNR = BURNR3
          ENDIF
  60     Q = BURNR*2.2046*9612./3600.
CC   THIS GIVES Q IN BTU/SEC
       ENDIF
   65 CONTINUE
       RETURN
       END


C
***  ***********************************************************************
       SUBROUTINE RADHT(T4WALL,VFMXC)
***  ***********************************************************************
       COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
       COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
       COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
       COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
       COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR

       DIMENSION VFMXC(579,579),T4WALL(579)
       DO 4010 K=3,NKM1
       DO 4010 I=2,NI
       II=(K-3)*(NI-1)+I-1
       T4WALL(II)=CONSRA*T(I,NJRA,K)*T(I,NJRA,K)*T(I,NJRA,K)*T(I,NJRA,K)
 4010 CONTINUE
C RADIATION FROM THE FIRE TO THE WALL
       DO 4011 J=3,9
       JJ=561+9-J
       AVT=0.25*(T(16,J,16)+T(17,J,16)+T(16,J,17)+T(17,J,17))
       T4WALL(JJ)=CONSRA*AVT*AVT*AVT*AVT
 4011 CONTINUE
C
       DO 4012 J=3,14
       JJ=568+J-3
       AVT=0.25*(T(6,J,16)+T(7,J,16)+T(6,J,17)+T(7,J,17))
       T4WALL(JJ)=CONSRA*AVT*AVT*AVT*AVT
 4012 CONTINUE
C
       DO 4020 I=1,560
       RWALL(I)=0.0
       DO 4020 J=1,579
       RWALL(I)=RWALL(I)+VFMXC(I,J)*T4WALL(J)
```

```
 4020 CONTINUE
      RETURN
      END


C
***   ************************************************************************
      SUBROUTINE GLOBE
***   ************************************************************************
*     THIS SUBROUTINE CALCULATES THE GLOBAL PRESSURE CORRECTION,            *
*     WHEREBY THE PRESSURE MATRIX IS UPDATED.                               *
*     VARIABLES USED ARE:                                                   *
*             SUMT        =      SUM OF TEMPERATURES                        *
*             SUMPT       =      SUM OF PRESSURE OVER TEMPERATURE           *
*             SUMPET      =      SUM OF EQUILIBRIUM PRESSURE OVER TEMP*
*             UGRT        =      CONSTANT                                   *
*             PCORR       =      PRESSURE CORRECTION                        *
***************************************************************************
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      SUMT=0.
      SUMPT=0.
      SUMPET=0.
      DO 370 I=2,NI
      DO 370 J=2,NJ
      DO 370 K=2,NK
      IF (NOD(I,J,K).EQ.1) GOTO 370
      DXI=XL(I,J,K,0,0,0)
      DYJ=YL(I,J,K,0,0,0)
      DZK=ZL(I,J,K,0,0,0)
      VOL=DXI*DYJ*DZK
      SUMT=SUMT+1./T(I,J,K)*VOL
      SUMPT=SUMPT+P(I,J,K)/T(I,J,K)*VOL
      SUMPET=SUMPET+REQ(I,J,K)*(1./1.0-1./T(I,J,K))*VOL
  370 CONTINUE
      SUMPET=SUMPET/UGRT
      PCORR=(SUMPET-SUMPT)/SUMT
      PCORRN=PCORR
      DO 371 I=1,NIP1
      DO 371 J=1,NJP1
      DO 371 K=1,NKP1
      P(I,J,K)=P(I,J,K)+PCORRN
  371 CONTINUE
      RETURN
      END


C
***   ************************************************************************
      SUBROUTINE SOLCON
***   ************************************************************************
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL22/ICHPB(10),NCHPI(10),JCHPB(10),NCHPJ(10),KCHPB(10),
     &           NCHPK(10),TCHP(10),CPS(10),CONS(10)
```

```
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      DO 402 N=1,NCHIP
      IB=ICHPB(N)
      IE=IB+NCHPI(N)-1
      JB=JCHPB(N)
      JE=JB+NCHPJ(N)-1
      KB=KCHPB(N)
      KE=KB+NCHPK(N)-1
      DO 405 I=IB,IE-1
      DO 405 J=JB,JE-1
      DO 405 K=KB,KE-1
      COND(I,J,K)=COND0*CONS(N)
      CPM(I,J,K)=CPS(N)
      NOD(I,J,K)=1
      IF (J.EQ.NJ) COND(I,NJP1,K)=COND(I,NJ,K)
      IF (I.EQ.2)  COND(1,J,K)=COND(2,J,K)
      IF (I.EQ.NI) COND(NIP1,J,K)=COND(NI,J,K)
      IF (I.EQ.2.AND.J.EQ.NJ)  COND(1,J+1,K)=COND(2,J,K)
      IF (I.EQ.NI.AND.J.EQ.NJ) COND(NIP1,J+1,K)=COND(NI,J,K)
      IF (J.EQ.NJ) CPM(I,NJP1,K)=CPM(I,NJ,K)
      IF (I.EQ.2)  CPM(1,J,K)=CPM(2,J,K)
      IF (I.EQ.NI) CPM(NIP1,J,K)=CPM(NI,J,K)
      IF (I.EQ.2.AND.J.EQ.NJ)  CPM(1,J+1,K)=CPM(2,J,K)
      IF (I.EQ.NI.AND.J.EQ.NJ) CPM(NIP1,J+1,K)=CPM(NI,J,K)
  405 CONTINUE
  402 CONTINUE
       RETURN
       END

C
***   ***********************************************************************
       SUBROUTINE PTRACK
***   ***********************************************************************
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
CC ** THE FOLLOWING PRESSURE TEST IS A TEMPORARY MEASURE TO MODIFY THE
CC    HEAT INPUT TO FORCE THE CALCULATED PRESSURE TO AGREE WITH THE
CC    EXPERIMENTAL PRESSURE.  IT WILL BE USED UNTIL ACCURATE HEAT INPUT
CC ** IS RECEIVED.
CC
      PSOUTH=P(10,9,16)*CONST1+REQ(10,9,16)
      PERROR=(PCURVE-PSOUTH)/PCURVE
      QCORR=1.0+PERROR-(PSOUTH-PM1)/PCURVE
      QCORR=1.0+PERROR-(PSOUTH-PM1)/PCURVE+(PSOUTH-PM1)/(PCURVE-PCURM1)*
     &    (PCURVE-PM1)/PCURVE
      QCORRT=QCORRT*QCORR
      PCURM1=PCURVE
      PM1=PSOUTH
C
      RETURN
      END


C
***   ***********************************************************************
      SUBROUTINE TCP
***   ***********************************************************************
*************************************************************************
*     THIS SUBROUTINE CALCULATES THE TEMPERATURE AT THE TERMOCOUPLE    *
```

```
*     POSITIONS.                                                          *
***************************************************************************
      COMMON/R4/XC(93),YC(93),ZC(93),XS(93),YS(93),ZS(93),
     &          DXXC(93),DYYC(93),DZZC(93),DXXS(93),DYYS(93),DZZS(93)
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)

      DO 5100 N=1,NTHCO
      II=NTH(N,1)
      JJ=NTH(N,2)
      KK=NTH(N,3)
      VOL=ABS((XC(II+1)-XC(II))*(YC(JJ+1)-YC(JJ))*(ZC(KK+1)-ZC(KK)))
      TCOUP(N)=0.
      DO 5101 I=II,II+1
      III=II+II+1-I
      DO 5101 J=JJ,JJ+1
      JJJ=JJ+JJ+1-J
      DO 5101 K=KK,KK+1
      KKK=KK+KK+1-K
      WVOL=ABS((XC(I)-CX(N))*(YC(J)-CY(N))*(ZC(K)-CZ(N)))/VOL
      TCOUP(N)=TCOUP(N)+WVOL*T(III,JJJ,KKK)
 5101 CONTINUE
      TCOUP(N)=TCOUP(N)*TR-273.18

 5100 CONTINUE
      RETURN
      END


C     ---------------------------------------------------------------------
***   *********************************************************************
      SUBROUTINE OUT(NN)
***   *********************************************************************
      COMMON/BL1/DX,DY,DZ,VOL,DTIME,VOLDT,THOT,TCOOL,PI,Q
      COMMON/BL7/NI,NIP1,NIM1,NJ,NJP1,NJM1,NK,NKP1,NKM1
     &  ,NIP2,NJP2,NKP2,NA,NAP1,NAM1,NB,NBP1,NBM1,KRUN,NCHIP,NJRA,NWRP
      COMMON/BL12/ NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,TINF,CNT,ABTURB,BTURB,VISL,VISMAX,QCORRT,PM1,PM2
      COMMON/BL16/ CONST1,CONST2,CONST3,CONST4,CONST6,NT,U0,H,UGRT,BUOY,
     &  CP0,PRT,COND0,VIS0,RHO0,HR,TR,TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR
      COMMON/BL32/ T(22,16,32),R(22,16,32),P(22,16,32)
     &       ,C(22,16,32),U(22,16,32),V(22,16,32),W(22,16,32)
      COMMON/BL34/ HEIGHT(22,16,32),REQ(22,16,32),
     &       SMP(22,16,32),SMPP(22,16,32),PP(22,16,32),
     &     DU(22,16,32),DV(22,16,32),DW(22,16,32)
      COMMON/BL37/ VIS(22,16,32),COND(22,16,32),NOD(22,16,32),RWALL(560)
     &       ,CPM(22,16,32),HSZ(3,2),NHSZ(22,16,32),RESORM(93)
      COMMON/BL38/NTHCO,CX(12),CY(12),CZ(12),NTH(12,3),TCOUP(12)
      COMMON/BL39/ALEW,PCURVE,CONSRA,PCURM1,PSOUTH,QCORR,PERROR
      XTIME=TIME*H/U0
      IF( NN .EQ. 1) THEN
C
      WRITE(6,500) XTIME,NTREAL,TIME,ITER,RESORM(ITER),SORSUM,Q
  500 FORMAT(1X, 'TIME=',F7.3,' S',1X,'NTREAL=',I9,1X,
     & 'TIME=',F7.2,'<0>',1X,'ITER=',I2,1X,'SOURCE=',
     & F9.6,1X,'SORSUM=',F9.6,1X,' Q(KW) = ',F10.4)
C
      QKW = ((60.*60.)/(3.412*1000.))* Q
      PRINT *
      PRINT *, '  PCURVE              PSOUTH                PERROR              Q
     &CRR            QCORRT              Q '
      PRINT *, PCURVE,PSOUTH,PERROR,QCORR,QCORRT,QKW
      PRINT *
C
      ELSE IF( NN .EQ. 2 ) THEN
      PRINT *
```

```
      PRINT *,'    TEMPERATURES  AT THERMOCOUPLE POSITION IN (C)'
      WRITE (6,*) (TCOUP(N),N=1,NTHCO)
      PRINT *
      PRINT *
      ELSE
      DO 502 L=16,16
      K=L
      DO 502 M=1,NIP1
      I=M
      WRITE(6,504) I,K
  504 FORMAT(/,2X,'I=',I2,5X,'K=',I2,/,10X,' T NOD',3X,'R(GM/C.C.)',2X,
     & 'U(CM/SEC)',2X,'V(CM/SEC)',2X,'W(CM/SEC)','P (ATM)',5X,'SMP',5X,
     & 'VIS(SEC/CM-CM)',3X,'COND(SEC/CM-CM)',' XSMP',/)
  513 DO 503 J=1,NJP1
C     XTEMP=T(I,J,K)/CONST3-273.16
      XTEMP=T(I,J,K)
C     XR=R(I,J,K)*RHO0/2.2048 *1000.*(0.0328)**3
      XR=R(I,J,K)
C     XU=U(I,J,K)*CONST6
C     XV=V(I,J,K)*CONST6
C     XW=W(I,J,K)*CONST6
C     XP=(P(I,J,K)*CONST1+REQ(I,J,K)*PINT)
      XP=P(I,J,K)
      XU=U(I,J,K)
      XV=V(I,J,K)
      XW=W(I,J,K+1)
CC    XVIS=VIS(I,J,K)*RHO0*CP0*H*U0*1.48814
CC    XCOND=COND(I,J,K)*RHO0*CP0*H*U0*1.48814
      XVIS=VIS(I,J,K)/VIS0
      XCOND=COND(I,J,K)/VIS0
      XSMP=SMPP(I,J,K)
      DDYY=1./FLOAT(NJM1-2)
      PE =SQRT(U(I,J,K)**2+V(I,J,K)**2+W(I,J,K)**2)*DDYY/COND(I,J,K)
      WRITE(6,511)J,XTEMP,XR,XU,XV,XW,XP,SMP(I,J,K),XVIS,XCOND,XSMP
  511 FORMAT(2X,'J=',I3,2X,F6.3,2X,F6.3,2X,F7.3,2X,F7.3,3X,F7.3,3X
     & ,F12.3,3X,F9.6,2X,F6.2,2X,F6.2,2X,F6.3)
  503 CONTINUE
  502 CONTINUE
      ENDIF
      RETURN
      END
```

LIST OF REFERENCES

1. Emmons, H.W. "Scientific Progress on Fire," Ann. Rev. Fluid Mech., Vol. 12, pp. 223-36, 1980.

2. Yang, K.T., J.R. Lloyd, A.M. Kanury, and K. Satoh, "Modeling of Turbulent Buoyant Flows in Aircraft Cabins," Combustion Science and Technology, Vol. 39, pp. 107-118, 1984.

3. Kou, H.S., K.T. Yang, and J.R. Lloyd, "Turbulent Buoyant Flow and Pressure Variations around an Aircraft Fuselage in a Cross Wind near the Ground," Fire Safety Science--Proceedings of the First International Symposium, pp. 173-184, 1986.

4. Nicolette, V.F., K.T. Yang and J.R. Lloyd, "Transient Cooling by Natural Convection in a Two-Dimensional Square Enclosure," Intl. J. Heat Transfer, Vol. 28, No. 9, pp. 1721-1732, 1985.

5. Aziz, K. and J.D. Hellums, "Numerical Solution of the Three-Dimensional Equation of Motion for Laminar Natural Convection," Physics of Fluid, Vol. 10, pp. 314-324, 1967.

6. Mallinson, G.D. and G. De Vahl Davis, "Three-Dimensional Natural Convection in a Box a Numerical Study," J. Fluid, Mech., Vol. 83, pp. 1-31, 1977.

7. Chan, A.M.C. and S. Banerjee, "Three-Dimensional Numerical Analysis of Transient Natural Convection in Rectangular Enclosure," J. Heat. Transfer, Vol. 101, pp. 114-119, 1979.

8. Ozeo, H., K. Fujii, N. Lior and S.W. Churchill, "Long Rolls Generated by Natural Convection in an Inclined, Rectangular Enclosure," Int. J. Heat Mass Transfer, Vol. 26, pp. 1427-1438, 1983.

9. Morrison, G.L. and V.G. Tran, "Laminar Flow Structure in Vertical Free Convection Cavities," Int. J. Heat Mass Transfer, Vol. 21, pp. 203-213, 1978.

10. Yang, H.Q., K.T. Yang and J.R. Lloyd, "Flow Transition in Laminar Flow in a Three-Dimensional Tilted Rectangular Enclosure," Heat Transfer, Vol. 4, 1495-1500, 1986.

11. Yang, H.W., K.T. Yang and J.R. Lloyd, "Laminar Natural Convection Flow Transition in Tilted Three-Dimensional Longitudinal Rectangular Enclosures," Int. J. Heat Mass Transfer, Vol. 30, No. 8, pp. 1637-1644, 1987.

12. Yang, H.Q., K.T. Yang and J.R. Lloyd, "Three-Dimensional Buoyant Bimodal Flow Transition in Tilted Enclosures," To appear in Int. J. Heat and Fluid Flow, 1987.

13. Yang, H.Q., Laminar Buoyant Flow Transitions in Three-Dimensional Tilted Rectangular Enclosures, Ph.D. Thesis, University of Notre Dame, 227 pp. 1987.

14. Ozoe, H., T. Shibata and S.W. Churchill, "Natural Convection in an Inclined Circular Cylindrical Annulus Heated and Cooled on its End Plates," Int. J. Heat Mass Transfer, Vol. 24, pp. 727-737, 1981.

15. Takata, Y., K. Fukuda, S. Hasegawa, H. Shimoomura and K. Sanokawa, "Three-Dimensional Natural Convection in a Porous Medium Enclosed in a Vertical Curved Annulus," Numerical Heat Transfer, pp. 29-39, 1983.

16. Takata, Y., K. Iwashige, K. Fukuda and S. Hasegawa, "Three-Dimensional Natural Convection in an Inclined Cylindrical Annulus, Int. J. Heat Mass Transfer, Vol. 27, pp. 747-754, 1984.

17. Rao, Y., Y. Miki, K. Fukuda and Y. Takata, "Flow Patterns of Natural Convection in Horizontal Cylindrical Annuli," Int. J. Heat Mass Transfer, Vol. 28, pp. 705-714, 1985.

18. Fusegi, T. and B. Farouk, "A Three-Dimensional Study of Natural Convection in the Annulus between Horizontal Concentric Cylinders," Proc. 8th Int. Heat Transfer Conf., San Francisco, California, Vol. 4, pp. 1575-1580, 1986.

19. Smutek, C., P. Bontoux, B. Roux, G.H. Schiroky, A.C. Hurford, F. Rosenberger and G. De Vahl Davis, "Three Dimensional Convection in Horizontal Cylinders. Numerical Solutions and Comparison with Experimental and Analytical Results," Numerical Heat Transfer, Vol. 8, pp. 613-631, 1985.

20. Yang, H.Q., K.T. Yang and J.R. Lloyd, "A Numerical Study of Three-Dimensional Laminar Natural Convection in a Horizontal Cylinder with Differentially-Heated End Walls at High Rayleigh Numbers," Proceeding of the Symposium on Heat and Mass Transfer Honoring Professor B.T. Chao,

University of Illinois, Urbana, Illinois, pp. 153-195, 1987.

21. Ozoe, H., K. Fujii, T. Shibata and S.W. Churchill, "Three-Dimensional Numerical Analysis of Natural Convection in a Spherical Annulus," Numerical Heat Transfer, Vol. 8, pp. 383-406, 1985.

22. Bagnaro, M., M. Laouisset, and F.C. Lockwood, "Field Model Prediction of Some Room Fires: Steady and Transient," Fire Dynamics and Heat Transfer, ASME HTD, Vol. 25, ASME, New York, pp. 107-114, 1983.

23. Markatos, N.C. and K.A. Pericleous, "An Investigation of Three Dimensional Fires in Enclosures," Fire Dynamics and Heat Transfer, ASME HTD, Vol. 25, ASME, New York, pp. 115-124, 1983.

24. Baum, H.R. and R.G. Rehm, "Calculations of Three Dimensional Buoyant Plumes in Enclosures," Combustion Science and Technology, Vol. 40, pp. 55-77, 1984.

25. Baum, H.R. and R.G. Rehm, "Natural Computation of Large Scale Fire-Induced Flows," Paper presented at the Eighth Int. Conf. on Numerical Methods in Fluid Dynamics, Aachen, West Germany, 28 June-2 July, 1982.

26. Baum, H.R. and R.G. Rehm, "Computation of Fire Induced Flow and Smoke Coagulation," Nineteenth Symposium, Intl of Combustion, The Combustion Institute, pp. 921-931, 1982.

27. Rehm, R.G. and H.R. Baum, "The Equations of Motion for Thermally Driven, Buoyant Flows," J. of Research of the National Bureau of Standards, Vol. 83, No. 3, pp. 297-308, May-June 1978.

28. Lloyd, J.R., K.T. Yang and K.V. Liu, "A Numerical Study of One Dimensional Surface, Gas, and Soot Radiation for Turbulent Buoyant Flows in Enclosures," Proceedings of the 1st National Conference of Math in Heat Transfer, University of Maryland, College Park, Maryland, pp. 142-161, 1979.

29. Yang, K.T., "Numerical Modeling of Natural Convection-Radiation Interactions in Enclosures," Proc. of 8th Int. Heat Transfer Conf., Vol. 1, pp. 131-140, 1986.

30. Alexander, J.I., H.J. St. Aubin, J.P. Stone, T.T. Street and F.W. Williams, "Large-Scale Pressurizable Fire Test Facility--Fire I," NRL Report 864, Naval Research Laboratory, Washington, D.C., December 1982.

31. Nies, G.F., Numerical Field Model Simulation of Full Scale Fire Tests in a Closed Vessel, Master's and Mechanical Engineer's Thesis, Naval Postgraduate School, Monterey, California, December 1986.

32. Eringn, A.C., Mechanics of Continua, John Wiley & Sons, Inc., 1967.

33. Boltz, R.E. and G.L. Tuve, editors, CRC Handbook of Tables for Applied Engineering Science, 2nd ed., CRC Press, Inc., Boca Raton, Florida, 1973.

34. Sparrow, E.M. and R.D. Cess, Radiation Heat Transfer, Hemisphere Publishing Corporation, Washington, D.C., 1978.

35. Department of Aerospace and Mechanical Engineering, University of Notre Dame, Technical Report TR-79002-78-2, "An Algebraic Turbulence Model for Buoyant Recirculating Flow," by V.W. Nee and V.K. Liu, 1978.

36. Patankar, S.V., Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Company, New York, 1980.

37. Department of Aerospace and Mechanical Engineering, University of Notre Dame, Technical Report, TR-37191-74-4, "A Numerical Model for the Prediction of Two Dimensional Unsteady Flows of Multicomponent Gases with Strong Buoyancy Effects and Recirculation," by Michael L. Doria, November 1974.

38. Leonard, B.P., "A Convective Stable, Third-Order Accurate Finite Difference Method for Steady Two-Dimensional Flow and Heat Transfer," Numerical Properties and Methodologies in Heat Transfer, ed. T.M. Shin, Hemisphere Pub. Corp., Washington, D.C., pp. 211-226, 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Mr. Stewart J. Marcus Code 0038 Naval Sea Systems Command Department of the Navy Washington, D.C. 20362	1
4.	Dr. Patricia A. Tatem Code 6183 Naval Research Laboratory Washington, D.C. 20375	2
5.	Professor K.T. Yang Department of Aerospace and Mechanical Engineering University of Notre Dame Notre Dame, Indiana 46556	3
6.	Professor M.D. Kelleher, Code 69Kk Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5004	2
7.	LT Janet K. Raycraft c/o Lucille Nisula Roucheleau Apt. 901 602 3rd Ave. No. Virginia, Minnesota 55792	2
8.	LCDR Richard Houck, Code 34 Naval Engineering Naval Postgraduate School Monterey, California 93943-5004	1
9.	Chairman, Department of Mechanical Engineering Code 69 Naval Postgraduate School Monterey, California 93943-5004	2

10. Professor Gilles Cantin, Code 69Ci 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93943-5004

11. CAPT Raymond Michelini 1
Code 56Y
Naval Sea Systems Command
Department of the Navy
Washington, D.C. 20362

12. Mr. James Gagorik 2
Code 211, ONT
Naval Sea Systems Command
Balston Towers #1
800 N. Quincy
Arlington, Virginia 22217

Numerical field model simulation of full scale fire tests in a closed spherical/cylindrical vessel.